FORMAL ONTOLOGIES MEET INDUSTRY

VISIT…

LANZAROTE Caliente.com

Frontiers in Artificial Intelligence and Applications

FAIA covers all aspects of theoretical and applied artificial intelligence research in the form of monographs, doctoral dissertations, textbooks, handbooks and proceedings volumes. The FAIA series contains several sub-series, including “Information Modelling and Knowledge Bases” and “Knowledge-Based Intelligent Engineering Systems”. It also includes the biennial ECAI, the European Conference on Artificial Intelligence, proceedings volumes, and other ECCAI – the European Coordinating Committee on Artificial Intelligence – sponsored publications. An editorial panel of internationally well-known scholars is appointed to provide a high quality selection.

Volume 174

Recently published in this series

Vol. 173. A. Holst et al. (Eds.), Tenth Scandinavian Conference on Artificial Intelligence – SCAI 2008
Vol. 172. Ph. Besnard et al. (Eds.), Computational Models of Argument – Proceedings of COMMA 2008
Vol. 171. P. Wang et al. (Eds.), Artificial General Intelligence 2008 – Proceedings of the First AGI Conference
Vol. 170. J.D. Velásquez and V. Palade, Adaptive Web Sites – A Knowledge Extraction from Web Data Approach
Vol. 169. C. Branki et al. (Eds.), Techniques and Applications for Mobile Commerce – Proceedings of TAMoCo 2008
Vol. 168. C. Riggelsen, Approximation Methods for Efficient Learning of Bayesian Networks
Vol. 167. P. Buitelaar and P. Cimiano (Eds.), Ontology Learning and Population: Bridging the Gap between Text and Knowledge
Vol. 166. H. Jaakkola, Y. Kiyoki and T. Tokuda (Eds.), Information Modelling and Knowledge Bases XIX
Vol. 165. A.R. Lodder and L. Mommers (Eds.), Legal Knowledge and Information Systems – JURIX 2007: The Twentieth Annual Conference
Vol. 164. J.C. Augusto and D. Shapiro (Eds.), Advances in Ambient Intelligence
Vol. 163. C. Angulo and L. Godo (Eds.), Artificial Intelligence Research and Development
Vol. 162. T. Hirashima et al. (Eds.), Supporting Learning Flow Through Integrative Technologies

ISSN 0922-6389

Formal Ontologies Meet Industry

Edited by

Stefano Borgo

Laboratory for Applied Ontology (LOA), ISTC-CNR, Trento

and

Leonardo Lesmo

Dipartimento di Informatica, Università di Torino

IOS Press

Amsterdam • Berlin • Oxford • Tokyo • Washington, DC

ISBN 978-1-58603-871-7
Library of Congress Control Number: 2008927475

Publisher
IOS Press
Nieuwe Hemweg 6B
1013 BG Amsterdam
Netherlands
fax: +31 20 687 0019
e-mail: order@iospress.nl

Distributor in the UK and Ireland
Gazelle Books Services Ltd.
White Cross Mills
Hightown
Lancaster LA1 4XS
United Kingdom
fax: +44 1524 63232
e-mail: sales@gazellebooks.co.uk

Distributor in the USA and Canada
IOS Press, Inc.
4502 Rachael Manor Drive
Fairfax, VA 22032
USA
fax: +1 703 323 3668
e-mail: iosbooks@iospress.com

PRINTED IN THE NETHERLANDS

Preface

This volume collects the papers that have been presented at the 3rd International Workshop on Formal Ontologies Meet Industry (FOMI 2008), that was held in Turin, Italy, on June 5th and 6th, 2008. FOMI is an international forum where academic researchers and industrial practitioners meet to analyze and discuss issues related to methods, theories, tools and applications based on formal ontologies.

There is today wide agreement that knowledge modeling and the semantic dimension of information plays an increasingly central role in networked economy: semantic-based applications are relevant in distributed systems such as networked organizations, organizational networks, and in distributed knowledge management. These knowledge models in industry aim to provide a framework for information and knowledge sharing, reliable information exchange, meaning negotiation and coordination between distinct organizations or among members of the same organization.

New tools and applications have been and are being developed in diverse application fields, ranging from business to medicine, from engineering to finance, from law to electronics. All these systems have exploited the theoretical results and the practical experience of previous work. In many cases, it has been shown that formal ontologies play a central role in describing in a common and understandable way the logical and practical features of the application domain.

The success of the methodologies associated with knowledge modeling and ontologies led to increased need of a comparison between different approaches and results, with the aim of evaluating the interdependencies between theories and methods of formal ontology and the activities, processes, and needs of enterprise organizations.

The FOMI 08 Workshop aims to advance in this direction by bringing together researchers and practitioners interested in ontology application, paying particular attention to the topics listed below.

- **Ontology and business**:
  ontology and ontological methodologies in business; adaptation of ontologies for companies and organizations; ontology effectiveness and evaluation in business;
- **Ontology and enterprise**:
  ontology-driven enterprise modeling; ontology development and change within organizations; ontology-driven representation of products, services, functionalities, design, processes;
- **Ontology and enterprise knowledge**:
  ontologies for the know-how; ontologies for corporate knowledge;
- **Ontology in practice**:
  ontologies for electronic catalogs, e-commerce, e-government; ontologies for mar-

keting; ontologies for finance; ontologies for engineering; ontologies for medical sciences;

- **Ontology and linguistics**:
  ontology-driven linguistic representation in organization knowledge; linguistic problems in standards and in codification processes; ontologies and multilingualism in business and organizations.

12 papers were selected for presentation at the conference. Beyond them, this volume also includes abstracts of the talks given by the invited speakers, who are Hans Akkermans, from the Vrije Universiteit of Amsterdam and Wernher Behrendt, from Salzburg Research Forschungsgesellschaft. We thank them for having accepted the invitation to participate in the workshop. We also thank all the members of the Program Committee for their work and for their suggestions. Finally, we would like to thank the members of the Local Organization Committee, Daniele Radicioni, Alessandro Mazzei, Cristina Bosco and Serena Villata: without their tireless activity the Workshop would not have been possible.

Stefano Borgo
Leonardo Lesmo

Conference Organization

Programme Chairs

Stefano Borgo
Leonardo Lesmo

Programme Committee

Bill Andersen
Francesco Bellomi
Guido Boella
Peter Clark
Matteo Cristani
Roberta Cuel
Roberta Ferrario
Michael Gruninger
Nicola Guarino
Gilles Kassel
Paulo Leitao
Riichiro Mizoguchi
Chris Partridge
Jorge Posada
York Sure
Valentina Tamma
Matthew West
Radboud G.F. Winkels

Local Organization

Cristina Bosco
Alessandro Mazzei
Daniele Radicioni
Serena Villata

External Reviewers

Valerio Genovese
Diego Magro
Alessandro Mazzei
Alessandro Oltramari
Daniele Radicioni

Contents

Formal Ontologies Meet Industry
S. Borgo and L. Lesmo (Eds.)
IOS Press, 2008

The Attractiveness of Foundational Ontologies in Industry

Stefano BORGO [a], Leonardo LESMO [b]
[a] *Laboratory for Applied Ontology (LOA), ISTC-CNR, Trento - borgo@loa-cnr.it*
[b] *Dipartimento di Informatica - Università di Torino - lesmo@di.unito.it*

Introduction

If we were to write a report on the ontology penetration rate in application-oriented domains like semantic web, database, engineering, business and medicine, just looking at the last year public events we would be justified in using an enthusiastic tone and even in going as far as to claim that ontology is nowadays a cornerstone in these areas. Indeed, many important conferences and specialized meetings devote considerable part of their time to ontology topics and are careful to register the new trends in ontological research. Prominent examples in 2007 are the OnTheMove Federated Conferences (OTM)[1] and the International Conference on Conceptual Modeling (ER)[2] in the database, business and infrastructure areas; the International Semantic Web Conference (ISWC), the Asian Semantic Web Conference (ASWC)[3] and the European Semantic Web Conference (ESWC)[4] in the domains related to Semantic Web; the International Joint Conference on Artificial Intelligence (IJCAI)[5], the Atlantic Web Intelligence Conference (AWIC)[6] and the Web Intelligence (WI)[7] in the vast Artificial Intelligence field. These are just a few of the international conferences that took place last year and explicitly related to ontology, not to mention the variety of associated workshops many of which are entirely dedicated to ontology, tools for ontologies and ontology application.

However, if we look closely at the data, we notice that these events are attended by companies and enterprises only in minimal part. Of course, one can refine this claim drawing several distinctions: between large and small/medium companies, between production industries and service providers, between developed and developing countries. Yet, there remains a feeling that the fuss about ontology is mainly at the level of research and its surrounding niches.

The Formal Ontologies Meet Industry workshop series began in 2005 to foster a positive relationship between formal ontology [1] and all the four sectors of industry: natural resources production (e.g., agriculture, fishing) and extraction (mining), manufacturing and construction, services to the business and consumers (from insurance and banking to education and health), and optimization research and design. From the experience gained in these appointments, it is clear that the gap between ontology research and industrial domain is wide and that the overlap between the two sides grows at a slow pace. We see several reasons for this, some of which have been discussed in the first FOMI report [2].

In this note we discuss another point, namely, the particular role of axiomatically rich theories based on formal semantics which are also known as *heavyweight ontologies*. In particular, among these systems we concentrate on foundational ontologies [3], that is, well designed and general heavyweight ontologies whose aim is to capture a clear perspective on reality by modeling philosophical positions (see e.g. [4]). The goal is to highlight the role of these sophisticated ontological systems in the industry domain and, vice versa, the role of industry (in the large) in ontological research.

Before entering into the discussion, it might be useful to point out that across the three FOMI editions, the majority of submissions are on the application of *lightweight ontologies*[8] to domains as far apart as chemistry, manufacturing, e-commerce, corporate knowledge, cultural heritage, network management and so on. This preference for lightweight ontologies (as opposed to heavyweight ontologies) is sometimes motivated but most of the time it is the result of a still imprecise understanding of the role of ontologies in information systems and of what one can actually do with the different ontological systems. In particular, if the layman does not always understand what distinguishes an ontological claim (any car has an engine, any car is always located somewhere) from an epistemic or factual assertion (any car has a radio device, any car can travel at 50 mph), we should not be surprised that she does not see the advantages of having part of the knowledge system deeply formalized. Similarly, we need some time before the novelties brought up by the new ontology discipline like, e.g., the formal distinction of properties, individual qualities and roles (this car *is gray* as requested, *the color of that hill* sticks out, this room *is an office*) can be fully appreciated in application fields. Naturally, the layman begins to take advantage of the new ontological perspective by exploiting it within well known techniques like, e.g., taxonomies. This attitude explains, at least in part, why today lightweight ontologies are popular in applications. Much harder is to understand the advantages of a new discipline and to exploit it in practice when the new perspective comes implemented into unfamiliar and sophisticated techniques, as in the case of heavyweight ontologies.

Lightweight ontologies, we said, do not bring new technical advantages due to their reliance on traditional approaches (classification techniques, graphical descriptions, glosses in pseudo-natural languages). Nonetheless, they witness the growing awareness of the importance of a correct terminology and of a careful (although necessarily informal) description of intended usage. The FOMI papers that are limited to lightweight ontologies are indeed quite interesting for the problems they bring up and the consequent discussions they arise.

Beside the variety of presentations based on lightweight ontologies, every FOMI event has seen some theoretical work and some example of heavyweight ontology application. Papers in the first group have been addressing the formalization of general notions, like function and product, and are helpful in forcing the practitioners and the researchers into (i) a clarification of the concepts that underlie these notion, (ii) a discussion of the variety of applications they can cover and (iii) the discovery of practical drawbacks that may be detected only working in real-world applications. More interesting is perhaps the work reported on the application of heavyweight ontologies since here the novelty of the ontological approach is seen in all its potentiality. These papers go to the core of the FOMI aims by providing information on new ontological methodologies,

like the formal application of descriptive theories (e.g., mereology) to model engineering scenarios within an ontological perspective, and by showing how these can be effectively implemented. Unfortunately, up to now, papers in this perspective have been submitted by academics only. In spite of the fact that these researchers work side by side with engineers and domain experts, this observation shows how the application of heavyweight ontologies still needs the leading role of theoreticians. That is, we lack tools and explicit methodologies to put domain practitioners in the position to independently experiment these systems.

The hope is that in the next few years we will see an increasing number of this latter type of works, perhaps with stronger participation of industrial personnel. For the moment, from the debate that accompanies the presentations at the workshop, we acknowledge that the practitioners in the domains spanned by FOMI are recognizing the role of rich and structured ontologies and show interest in their potentialities. Of course, it takes more than these few papers at FOMI to push them into an active investigation of the richness of ontology research: the area has to gain in stability, clearness and maturity.

The Role of Foundational Ontologies in Industries

Nowadays a few foundational ontologies are available and are being tested in different domains and application projects. The initial hope to reach a unique general comprehensive ontology (or ontological framework) that unifies all ontological perspectives is definitely abandoned. People that see ontology as a tool to make systems or applications interoperable, find discouraging that foundational ontologies themselves suffer from the interoperability problem. If we need to rely on them – they argue – it seems we should first find a way to integrate or make them interoperate. However, this is effectively a complex issue and, notwithstanding some results, one may think that it is better to give up on foundational ontology and develop instead direct mappings between the numerous systems and modeling techniques that are in practice today. To put things straight on this issue, we have two different questions to address, namely:

- Given that there are several distinct foundational ontologies, is this a problem for interoperability?
- What is a foundational ontology good for in industry?

Regarding the first question, our answer is ‘no’. The view of foundational ontology as the glue that allows us to assemble different applications into a unique coherent and interactive system is quite naïve. What these ontologies do is to formalize an explicit “view on reality” by clearly indicating what is assumed to exist and how things are assumed to relate to each other. The goal, thus, is far from having everything under one single description of reality. The idea is that, in order to reliably communicate and interact, one needs to know what others (people, agents, organizations or artificial systems) believe about reality and, it is assumed, this result is effectively achieved once one has available the foundational ontology that best captures their view of reality. That is, to be able to interact with another system, we need to have available its foundational ontology. In this way one can build a formal interface to translate information from one system to another by coding the information in the first into the view of reality that the latter adopts.

Regarding the second question, we answer with an example.

Several authors in the engineering domain have been working on the notions of function and behavior from different perspectives: artificial intelligence, system modeling, product description, and so on. B. Chandrasekaran and J.R. Josephson in [5] claim that, relatively to the areas they considered, engineers use five different meanings for the term 'behavior' and two for the term 'function' and informally discuss the relationship among these behavior(s) and function(s). To take advantage of this result, one should be able to formalize and logically relate these different meanings. This has been shown to be possible via a foundational ontology [6]. The role of foundational ontology in this work is crucial: it motivates and provides the general framework in which it becomes possible to model the different meanings of these general terms. It is important to note that in this "ontologization" the meanings given by Chandrasekaran and Josephson have been modeled without discussing their value (if engineers say these meanings are what they need, one should capture exactly and precisely those meanings) nor trying to twist them into the ontology framework (either they fit the ontology view of the world or we need a different ontology). Technically, the result is expressed in a series of logical formulas that show how these different meanings depend on an ontological notion of behavior (and function) and how they are related to each other while remaining different.

Nonetheless, one may still claim that a foundational ontology is not really necessary to reach interoperability. Perhaps, one could insist, just a set of general concepts for design and manufacturing, concepts like functionality, product, process, production plan and so on, is enough for this domain without any need to refer to a foundational system. We claim that this dismissing position underestimates the variety of the industrial domain. Indeed, a closer analysis of the case described above shows that the notions of function in Chandrasekaran and Josephson depend on the notions of behavior. However, this is not true in other approaches. The Functional Representation approach of Chandrasekaran and Josephson is just one among several that rely on these very notions like the Function-Behavior-Structure of J. G. Gero [7,8], the Function-Behavior-State of Y. Umeda [9], the Structure-Behavior-Function by A. Goel [10,11]. Instead, R. Stone and K. Wood do not even make use of the notion of behavior in their Functional Basis model [12]. In some cases the relationship between function and behavior is reversed: the very notion of behavior is seen as a specialization of that of function. For a different example, one may hope to get a shared view on specific notions like 'shape' or 'electric power', but for fairly general concepts, like operation in the shop floor, we have to face overt ontological issues if we want to be able to use such a notion together with those of plans, processes and agents' actions.

The different approaches in the various industrial domains rely on different (often implicit) local ontologies and primitives; the problem of relating these is inherently ontological. Once we have clarified and organized the different meanings of these key concepts, we can further specialize them to coherently capture specific terminology in the different applications and representation systems.

What is Ontological Analysis Good for in Industry?

The introduction of ontological analysis [13] in artificial intelligence and knowledge modeling is motivated by several considerations: the limit of domain dependent model-

ing, the failure of interoperability among independently developed systems, the analysis and classification of background assumptions etc. These issues are important in the modeling of the enterprise as well as of the production process and affect the potentialities of the modern industry domain, from the exploitation of virtual enterprises to the development of integrated product lifecycles. The answer provided by the research in ontology is orthogonal to (and integrates with) the innovative techniques from other disciplines: it aims at deepening and make transparent our knowledge of the systems and of the environment they live in. For example, it clarifies the different uses of the term 'product' within the same automotive company or across the supply chain, it identifies the functionalities that are potentially realizable by services or embedded by technical artifacts, it provides uniform ways to model properties and to translate them in formats intelligible to different legacy systems.

One should not think that the aim of ontological analysis is to build good ontologies only. It helps to improve existing systems like standard databases. The ontology and the database communities yield, at a minimum, different perspectives: the purpose of ontology is to define and classify *categories* of entities (classes like *Drilling machines* as opposed to a specific instance like *The drilling machine item #123*) mainly by organizing and relating their formal properties and interactions. The purpose of databases is to collect information to describe *situations* by representing entities (classes as well as instances) and their actual relationships. The goal is the completeness (and efficient management) of the information which is deemed necessary in some domain or application, not the distinction between its ontological or factual nature. For example, an ontology must distinguish processed items from items on sale, although the two classes of entities may coincide in a given company, while a database may intentionally ignore this conceptual distinction in order to simplify the repository and improve data quality. Ontological analysis is today used to improve traditional databases as well as to create optimal and transparent interfaces for their interoperability. The role of ontologies in this latter area is, perhaps, fairly well understood. The first task consists in developing sound modeling guidelines and improving the use of well established representation languages. An example can be found in [14] where modeling constructs of attributes and datatypes are analyzed and a methodology is developed for the UML modeling language.

The Role of Industry in Foundational Ontology

Foundational ontologies are not tools for every application. For example, several applications in the Semantic Web can safely rely on lightweight ontologies because they strongly depend on statistical analysis of large sets of data, e.g., classifications based on trends or social networking, or because they explicitly give up on precision and clearness in favor of other aspects like simplicity of use, emotional descriptions, personal web, ephemeral classes and descriptions. Foundational ontologies are sophisticated and expensive to produce; although their use can be valuable in any domain (including the classification of trends and personal tagging), their construction and adoption should be motivated by a cost-benefit analysis. Domains based on sophisticated artifacts that have a relatively long life cycle and whose construction, maintenance and update requires careful analysis and considerations of aspects like functionality, requirements, implementation and sustain-

ability (airplanes, radar systems, civil constructions, electric or oil networks, biomedical instruments and so on), find great advantages from a principled approach based on foundational ontologies. The reason lies in the real possibility, brought by these ontologies, of integrating the information across the whole industrial process: from the design phase to the after-sale services, from product update to manufacturing adaptation, functionality control and guidebook update. These observations show the interest in exploiting foundational ontologies in domains like industry and medicine beside the traditional area of the Semantic Web.

The advantages brought by foundational ontologies are easily foreseen but the deployment of these ontologies in complex domains cannot take place without some initial investment. Foundational ontologies require time to develop and in this phase in which just a few of these ontologies are available and their exploitation is just at a start, industries should invest considerable resources to build, refine or adapt these ontologies to their needs. Even more challengingly, these ontologies are too sophisticated to be used or understood by untrained people: the development of appropriate tools by which the average employee can effectively use them, perhaps relying on a training period of a few hours only, would require considerable efforts. These are real drawbacks and should be openly faced in order to understand the real industrial needs and to address investments in the right direction. Right now, most projects that rely on ontology in this area are still based on public funding and only in some rare cases are supported or initiated by industrial consortia (the example here is the EPISTLE Core Model[9], an ISO standard which has been recently proposed as a top-level ontology).

Since foundational ontologies are, after all, new tools for industrial needs, the development of an ontology as an open or proprietary standard depends on industrial considerations. However, if we want to take advantage of web technology and the new approaches toward virtual enterprises and integrated supply chains (just to name a few cases), we see that these general ontologies will have major impact and will provide the biggest advantages if they are publicly shared and widely adopted, or at least widely recognized: due to the costs of restructuring industrial information systems, many enterprises may at first prefer to enhance their legacy or proprietary systems by providing an interface that aligns the enterprise data and knowledge structures to a standard ontology. In this way, an enterprise can take advantage of a standard ontology from the beginning and avoid to redesign at once the whole information system to conform to the ontology: a change that requires investment in terms of money, time, and personnel training.

It is sometime claimed that foundational ontologies are not suited if the target domain is quickly evolving as it happens, e.g., in software and artifacts based on new technologies. The argument, as far as we see, relies on the confusion between types of ontologies (foundational, core, domain, formal, lightweight, upper level etc.) and knowledge bases (or even databases). Foundational ontology sets the knowledge structure by establishing the meaning of the concepts central to a given domain, thus it *defines* the very domain at stake: in the industry domain a foundational ontology would be extended to formally represent concepts like artifact, component, feature, function, process, service, operation, agent, and so on. The evolving set of products and product models or the specific functionality of an item are pieces of information that one finds in knowledge bases, not in foundational ontologies, and knowledge bases have long proved that they can cope with evolving environments, provided they are well constructed and maintained. Ontolo-

gies can help in providing the correct framework on top of which to construct optimal knowledge bases and databases for the domain perspectives and needs, which includes the capability to model new product types and to construct and discover new functionality types. In principle, a foundational ontology is quite stable over time if we exclude possible extensions to include new general concepts.[10] Companies should learn to distinguish the different types of ontologies since these are developed to answer different types of problems. In particular, they should be able to distinguish between foundational ontologies and knowledge bases; two complementary systems whose alignment is crucial for the success of the evolution toward the ontology-based enterprise.

Finally, industry should be less shy in addressing the research community regarding ontology and ontology applications. The gap between ontology research and ontology implementation has brought many researchers to spend most of their efforts toward ontology languages and reasoning classifications. The consequence is that crucial aspects to improve usability [15] like terminology development (needed to foster understanding and correct usage by non-experts) and ontology interfaces for the end-user [16], do not hold the stage today. Terminology development and ontology interface development are of course just two aspects of the pervasive relationship between ontology, natural language and human/computer interaction, a relationship which today has to be assessed with respect to the main source of information: the Web.

Ontology, Language and Communication

It is well known that the vast majority of information sources is based on natural language. Most Web pages include portions of text written in one of the hundreds of languages of the world. The next step in the process of making Web information available to an even larger set of users is to provide for access in written or spoken language, possibly exploiting the current technology for dialogue management. But this can hardly be achieved without an in-depth shared view on the meaning of the individual words.[11]

Unfortunately, also in this field, the huge number of technical and scientific papers describing possible relations between language and ontologies (ranging from the use of ontologies for understanding texts to the use of texts for building or extending ontologies) is not mirrored in a comparable number of real-world applications. Nonetheless, some software products for specific fields of application do exist; the most relevant example concerns perhaps healthcare (`http://www.landcglobal.com/`). The limited diffusion of NLP tools exploiting (lightweight) ontological knowledge can be taken as supporting two claims: ontologies are useful for language analysis, but their development is highly expensive (requiring consistent fundings, as are more easily available in the medical field).

The first point is hard to deny: the extraction of the meaning of a piece of text involves the representation of this meaning in some suitable “internal language” enabling a machine to perform various types of reasoning. Of course, this is not needed if what is needed is a link to a web page (but the current techniques for Web lookup can hardly be said to involve meaning extraction) or a summary (where the representation language is the same as the original text). But if an internal language is needed, it must have some

formal flavour; usually, it will be a logic language, including predicates and constants.[12] If we want to move toward the sharing of this *meaning* among systems, there must be an agreement about the predicates used and about what they mean. This can be achieved (at least in part) just via an agreement on the meaning of terms, as the one provided by the adoption of shared lightweight ontologies.

The second point, i.e. the high cost incurred in the development of ontologies, is not less agreed upon. The obvious question that an industry or a public administration asks itself is: is it worth? Though this question and the possible answers have been analyzed in depth in the previous sections, something should be added here. Much effort is being devoted to the automatic development of ontologies, especially lightweight ones. Usually, the input data for this development are pieces of text, but the current status of these automatic techniques yield results that are only partially encouraging. Again, it seems that there is no way out, especially in the case one wants to face the problems associated with the "ephemerality" of classes mentioned in the previous section. Arguably, more effort is required on this topic, possibly adopting approaches (that are based on a sort of *bootstrapping*) where language is analysed in more depth, on the basis of established ontologies where the meaning of terms is reasonably fixed and shared. However, it is hard to imagine that these efforts can achieve their goal in a short time without a strict cooperation between industry and academy.

But language also teaches us a useful lesson: if I receive a "Call for participation to the FOMI workshop", I am usually able to understand what it means, i.e. what is a *workshop*, what would involve for me to *participate*, what the *call* has been sent for. The same probably holds for my friends and colleagues in Japan, Germany, India, South Africa, etc.. Concepts as *workshop* (a complex event) or *participation* (perhaps an activity) are notoriously difficult to model in a heavyweight ontology and receive different formalizations in different approaches. Nonetheless, people growing up in different cultures, speaking different languages and having different life experiences, are successful to talk to each other and to *understand* each other. One may ask: Are they using the same or different ontologies? Is there a common core that is *common* in the true sense of this word? Can some form of ontology capture this core? What combination of semantic interaction components should we look for?

Having acknowledged in this paper the existence and value of different ontological approaches and their different goals, let us close with a word of hope: maybe not in the next, but in some FOMI workshop close in the future, we will discuss about different ontologies acting as the basis for real interoperation: interoperation between humans and humans (via web pages written in different languages), between humans and machines (where we will be able to give commands to robots in such a way that we and the robot have a common understanding about what a command or a required operation is), and interoperation between computer systems. At that point, we will be able to say that the final goal of ontological studies for improving everyday life and industrial applications is being approached.

Notes

[1]`http://www.cs.rmit.edu.au/fedconf/`
[2]`http://er2007.massey.ac.nz/`

[3]http://iswc2007.semanticweb.org/main/default.asp
[4]http://www.eswc2007.org/programmecommittee.cfm
[5]http://www.ijcai-07.org/
[6]http://www.awic2007.net/call-for-papers
[7]http://www.cs.sjsu.edu/wi07/
[8]The term is generally applied informally. Here we use it to refer to semantically weak systems like those based on taxonomies, concept maps, or conceptual schemata.
[9]See the "Industrial automation systems and integration – Integration of life-cycle data for process plants including oil and gas production facilities – Part 2: Data model"; it can be accessed at: http://www.tc184-sc4.org/wg3ndocs/wg3n1328/lifecycle_integration_schema.html.
[10]Think, e.g., about the new concepts brought into the manufacturing based economy by the globalization and information technology.
[11]"In order to perform the kind of reasoning/inference required for deeper (semantic) understanding of texts, as required for high-quality Machine Translation, Summarization, and Information Retrieval, it is imperative to provide systems with a wide-ranging semantic thesaurus."(Objectives of the SENSUS Project http://www.isi.edu/natural-language/projects/ONTOLOGIES.html)
[12]If one gets a DB query, then it will include names of relations and fields and values appearing therein.

References

[1] N. Guarino. Formal ontology in information systems. In N. Guarino, editor, *Proceedings of the Second International Conference on Formal Ontology in Information Systems*, pages 3–15. IOS Press, 1998.

[2] Stefano Borgo, Matteo Cristani, and Roberta Cuel. Formal Ontology meets Industry. *Applied Ontology*, 1(3-4):217–220, 2006.

[3] Stefano Borgo. How Formal Ontology can help Civil Engineers. In Jacques Teller, John Lee, and Catherine Roussey, editors, *Ontologies for Urban Development*, Studies in Computational Intelligence, pages 37–45. Springer, 2007.

[4] C. Masolo, A. Gangemi, N. Guarino, A. Oltramari, and L. Schneider. Wonderweb deliverable d17: The wonderweb library of foundational ontologies. Technical report, 2002.

[5] B. Chandrasekaran and J.R. Josephson. Function in device representation. *Engineering with Computers*, 16(3/4):162–177, 2000.

[6] Stefano Borgo, Massimiliano Carrara, Pawel Garbacz, and Pieter E. Vermaas. A Formal Ontological Perspective on the Behaviors and Functions of Technical Artifacts. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 23(1):(to appear), 2009.

[7] J.S. Gero. A knowledge representation schema for design. *AI Magazine*, 11(4):26–36, 1990.

[8] J.S. Gero and U. Kannengiesser. The situated Function-Behavior-Structure framework. *Design Studies*, 25:373–391, 2004.

[9] Y. Umeda, M. Ishii, M. Yoshioka, Y. Shimomura, and T. Tomiyama. Supporting conceptual design based on the function-behavior-state modeler. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 10:275–288, 1996.

[10] A. Goel. A model-based approach to case adaptation. In *Proc. 13 Ann. Cogn. Sci. Conference*, pages 143–148. Mahwah, NJ: Lawrence Erlbaum, August 1991.

[11] S. Bhatta and A. Goel. Design patterns and creative design. pages 271–284, Berlin, 2002. Springer.

[12] R.B. Stone and K. Wood. Development of a functional basis for design. *Journal of Mechanical Design*, 122(4):359–276, 2000.

[13] Nicola Guarino. Formal ontology, conceptual analysis and knowledge representation. *International Journal of Human and Computer Studies*, 43(5/6):625–640, 1995.

[14] Giancarlo Guizzardi, Claudio Masolo, and Stefano Borgo. In defence of a trope-based ontology for conceptual modeling: An example with the foundations of attributes, weak entities an datatypes. In *Conceptual Modeling - ER'06*, volume LNCS 4215, pages 112–125, Tucson, Arizona (USA), 2006. Springer.

[15] Mark S. Fox and Michael Gruninger. On ontologies and enterprise modeling. In *Proc. Int'l Conf. Enterprise Integration Modeling Technology*. Springer-Verlag, 1997.

[16] Martin Hepp. Possible ontologies: How reality constrains the development of relevant ontologies. *IEEE Internet Computing*, 11(1):90–96, 2007.

Formal Ontologies Meet Industry
S. Borgo and L. Lesmo (Eds.)
IOS Press, 2008

The Business of Ontology calls for a Formal Pragmatics

Hans AKKERMANS

The Network Institute
Informatics Department
VU University Amsterdam
www.thenetworkinstitute.eu; www.e3value.com

Abstract. I develop a few suggestions how formal ontology can meet industry and practice. This is not just a matter of useful ontology-based applications. That is a necessary but also rather trivial idea that perpetuates the gap between fundamental and applied research and keeps alive associated outdated linear process ideas of innovation. Instead, I suggest that for further progress the ontology field is to move from a semantic to a pragmatic approach in the communication-theoretic sense. Formal pragmatics stands a better chance to provide a rational and scientific foundation for an integrated ontology theory as well as practice than the deductivist formal semantics approach.

Introduction

The research tradition that goes under the name of "ontology" within Computer Science (CS) and, more specifically, Information and Knowledge Systems (IKS) is now about twenty years old. Whatever the adopted measure – strictly academic or in terms of socio-economic or industrial usefulness – the field of ontology has been extremely successful, much more so than the first generation of CS "ontologists" and "ontology engineers" (including the present author) had anticipated.

After twenty years, it is a good time to take stock, not only of what has been achieved, but (more importantly and interestingly) also what still needs to be done, and especially what has to be changed in the direction we are going if we undertake to make further significant scientific as well as practical-applicational progress. In the following I develop (continuing work in [1]) a few observations, positions, and proposals on this.

1. Ontologies are us (or them?)

To cite the very recent (December 2007) definition in the Handbook of Knowledge Representation [2]:

> "An ontology is an explicit specification of a shared conceptualization that holds in a particular context" [2].

The (slow and gradual) changes of the definition of what an ontology is are interesting in their own right. Gruber's original definition takes ontology to be an explicit specification of a conceptualization [3]. The introduction of the adjective *shared* occurred some years later, especially in the work of Borst et al. [4]. The reference to validity in a particular *context* is again a later addition, and attempts to encapsulate lessons learned in building and using ontologies in practical applications, especially in knowledge management [5] and on the Web [6].

Increasingly, the realization now is that context first of all concerns *social* context. Ontologies as conceptualizations express social constructions and play a role as such in Web communities of practice and social networks, an idea aptly formulated by Péter Mika as "ontologies are us" [7]. Thus, the original idea of ontologies is that of a consciously engineered artefact to express intended (and as far as possible universalistic) meaning through specifying the formal semantics of concepts. In the latter conception, ontologies figure as social network-created, even emergent, "meaning productions" inseparable from the context of the community in which they are created.

Many authors have noted that these two different approaches to ontologies do not necessarily exclude each other. There are many different types of ontologies that have demonstrated practical usefulness in applications, in both approaches. Also, both approaches need critical scrutiny. In the engineering approach, an important critical issue is the distance (often expressed in terms of intuitiveness and perceived complexity) between the engineered ontology and its intended user or application community. In the social community approach, on the other hand, a critical issue is that in ontology extracted as a matter-of-fact empirical (i.e. non-engineered) social phenomenon ultimately "anything goes": shared meaning simply becomes what any community happens to believe, without any further rational foundation or justification. Just one example I am thinking specifically of is the xenophobia that has in recent years become a plague in my own country and others, and that also has found its way to the Internet in blog communities. It would be an easy task to extract a really shared ontology here (certainly when applying the maxim or slogan [2] "smaller ontologies are better"), but it will be one that is much less harmless than the conceptualizations shared by, say, the Semantic Web research community. Artefacts should not be approached uncritically, but neither should communities. Rational foundations and justifications continue to be necessary in both cases.

2. Formal pragmatics (rather than formal semantics)

Nevertheless, a fundamental realization that is here to stay is (i) theoretically, the inherently *social* nature of ontologies as shared conceptualizations; (ii) practically, the inherently *contextual* nature of ontologies as useful and valid support tools for information and knowledge sharing. So, as a next step, ontology as a scientific and computational engineering theory of conceptualization needs to account for this in a principled way.

A starting point to do so (cf. [8]) is to reconsider the, underlying and in part implicit, conception of *meaning* itself as it has been employed e.g. in ontology engineering. To date, much of ontology engineering derives its foundational ideas from the standard truth-conditional view: essentially, the meaning of a sentence or

proposition or utterance coincides with the conditions for its truth. This is a view that goes back to logicians such as Frege and Tarski and is still held by many modern model-theoretic logicians, formal semanticists, theoretical computer scientists, and knowledge representation researchers.

However, the above characterization of ontology as inherently social and contextual in nature leads us to entertain a different foundational notion of meaning: the meaning of a sentence or proposition or utterance lies in its actual *use* in communication. This is a view that goes back to the later Wittgenstein, and has been developed by a (wide) variety of philosophers such as Dummett, Grice, Austin, Searle, Habermas, linguists such as Sperber and Wilson, and argumentation theorists such as Van Eemeren & Grootendorst, and Walton.

I contend that this pragmatic view of meaning will show to be the more fruitful one for the field of ontology engineering and the Social Semantic Web. Rather than formal semantics, in ontology we need a both conceptual and computational theory of formal pragmatics. To come to such a theory, I further contend that there are several research lines, also in computer science and artificial intelligence, with relevant results already available that are just waiting to be taken up. Below, I will suggest just a few components of this.

3. Problem-Solving Methods: what happened to them?

As quite an elementary point of context, ontologies are formal conceptualizations not made *l'art pour l'art*, but to help achieve a goal or task by an actor. Often that task involves knowledge-intensive reasoning. The conceptual distinctions that we make in our attempts to understand the world are not just static and descriptive domain characterizations, but they are made to serve practical purposes of action by someone or something in that world. We cannot detach knowledge from action.

Task and actor characteristics are key parts of operational context definition [9], but left out or kept implicit in much current Web ontology research.

Two decades of knowledge engineering (cf. [2] and references therein) have delivered a wealth of evidence that there are recurring patterns or stereotypes in the structuring and use of knowledge as an instrument in tasks that involve reasoning and computing. These recurring knowledge stereotypes are variously referred to as inference schemas, task templates, strategy patterns, or problem-solving methods (PSMs). PSMs are heuristic and stereotypical in the sense that they do not guarantee to solve a given knowledge-intensive problem in general (unlike a normal algorithm). But, they do have demonstrated pragmatic value in solving common, typical, or average cases of knowledge-intensive tasks that can, moreover, be reused in many different situations. The 80-20 rule applies here: that PSMs as stereotypical schemas only handle the typical or average case is often seen as a kind of algorithm bug in traditional computer science, but it is in fact a useful feature on the Web as an open dynamic system of unknown but certainly very high complexity (cf. [10]).

As an example, consider PSMs such as Propose-and-Revise or Propose-Critique-Modify that have been originally developed for configuration design tasks in engineering. Such tasks have customer requirements as input, and an artefact

description as output. The specific task type of configuration design presupposes a generic domain ontology in which predefined functional components, "hard" constraints, and "soft" partial preference orderings are key concepts, on top of which the PSMs can do their work. Concepts from the domain ontology are then invoked by the PSM as a specific knowledge role (i.e. way of use) in a certain phase of the reasoning process.

This is clearly a pragmatic, use-oriented view of ontologies. Part of the context surrounding the normal static ontologies is computationally defined by explicating the reasoning task and methods in which ontologies find their use. Another part of context is explicating the task from the (often non-technical) perspective of the agent or actor who carries out that task, such as users or customers. In the family of networked business ontologies (such as e^3*value*) that our Business Informatics group at Amsterdam has developed, we employ configuration PSMs for automated service bundling, based on a service ontology that does not take – as is usually the case – the technical Web Service perspective, but defines services as an economic/commercial entity in business supplier-customer terms [11]. Other PSMs (planning, parametric design) are also candidates for provisioning of service bundles over the Web.

Two decades of knowledge engineering has provided a library of useful task decomposition and inference method schemas. Semantic Web and Web Service research has tended to overlook much of older PSM research, with the result that handling different task and actor contexts is dealt with in an unnecessarily limited (i.e. unintelligent) way. Instead, and on their turn, PSMs could themselves be made available as services on the Web. For that to work, they also need to be supplied with background knowledge how to use them as a service, i.e. with some kind of task/method ontology that explicates in what kind of context (typical or average case) they work and what the requirements (knowledge roles of domain ontology) for their practically successful functioning are.

4. The dialectical turn: argumentation in dialogue

Another relevant strand of research in pragmatics that has recently made great theoretical and computational progress is argumentation theory (for a survey, see [12]) and the related fields of informal logic and critical thinking. Analogous to PSMs, it has produced a sizable library of argument schemas that model often occurring prototypical arguments. It has two important characteristics relevant to ontology-based reasoning and semantic approaches to the Web (even though it does not have much to say about the issue of conceptualization *per se*):

- It breaks away from the deductivism that is still prevalent also in formal ontology and semantics. Namely, there are many arguments and argument schemas that are not deductively valid, are defeasible instead, but still are to be considered as good and acceptable (typically because they withstand scrutiny by thorough critical questioning). These non-deductive, "presumptive" arguments naturally occur in settings of commonsense everyday reasoning, are irreducible to deductive reasoning, but still can be explicated in a formalized and computational way.

- The dialectical turn. It views reason and argument as an inherently social process: argumentation concerns a certain type of (rational) communication and dialogue that is subject to certain social rules and principles. Critical questioning is the dialectical mechanism that tests the quality of an argument and its claim in contexts where deductive certainty is impossible to come by (because of bounded resources, time, lack of information, etc.). Clearly, this is the context that constitutes the normal condition of the Web.

As an example, argumentation theory and informal logic have developed reusable schemas for practical reasoning (answering the question: what should we do if we want to achieve goal G, under conditions of uncertainty). That has some promise to be applicable when reasoning about service provisioning over the open dynamic system environment of the Web. One might consider constructing this as a kind of PSMs with built-in critical tests for their applicability and validity.

Very first attempts to introduce an argumentative approach to the Semantic Web go under the flag of the claim web [13] and the argument web [14]. Independently of the value of such proposals, however, argumentation theory has several important things to say in general to the field of ontology engineering and the semantic side of the Web.

As a theory of (a certain area of) pragmatics, argumentation theory is able to cover several aspects of context, especially concerning the communicative relationships between actors. This is definitely of importance in talking about a Social Semantic Web. It furthermore introduces ontological distinctions between different types of dialogue, which in fact provides a characterization of different kinds of communicative goals and tasks of actors. Rather than a purely empirical emergent semantics [15] based e.g. on gossip algorithms as a mechanism modelling communication between agents, it could provide more structured and critically-rationally justifiable alternative mechanisms, in the form of dialectical-argumentative schemas which are still social and communicative in nature. Thus it not only widens the range of possible mechanisms to achieve forms of self-organization (more generally, self-* properties) on the Web, but also might give them a better rational foundation that is moreover communicatively sharable.

5. Formal Pragmatics and discursive rationality

A natural extension of my proposed idea to introduce results of theories of pragmatics into ontology engineering and semantic-based approaches to the Web, concerns various brands of speech act and communicative action theory [16, 17]. In a basic form, Searle's speech act theory has found its way into agent communication languages. But, in current semantic approaches to ontology-based Web (service) engineering this plays at most a peripheral role.

However, speech act theory is important to the Web because it extends – much more radically than ontology-based PSMs and argumentation schemas – the types of communicative acts that can be treated computationally. Formal semantics, also in ontology engineering, is still very much focused on those utterances that express propositions or assertions (true or false beliefs about the world). But there are many other types of speech acts that play an important role in an open Web environment that

is of a social nature. In e-business, for example, the majority of speech acts relates to negotiating and transacting and so is definitely not of a propositional nature. This is a phenomenon not adequately accounted for in current Web-semantic based approaches. In a semantic-based Service Science [18] and ICT-based innovation that is to be practically as well as scientifically adequate the non-propositional character of much Web communication cannot stay in the periphery of attention, as is now in fact the case. However, as indicated above, by employing and integrating various strands of existing research results relevant to the pragmatics of the Web and of Web intelligence, significant progress is in my opinion clearly within reach.

Further developments related to speech act, communication and social network theory have taken place but are still beyond the horizon of ontology and semantic-based technology. For example, Searle in his critique of the standard model of rationality in [16] argues that the intentions, desires, and goals of agents are not input to reasoning but should be part of reasoning. Habermas's communicative action theory sees *all* speech acts (not just propositions that assert something) as inherently making validity claims (regarding relevance, truth, rightness, truthfulness) that are *discursively* criticizable; and in this fact also lies their rational justification [17]. If we develop a corresponding knowledge-level computational framework for this, we will have not a Web of data, not a Web of enriched semantic annotations or of community chat and gossip, but a *reflective* Web. Truly reflective open systems are currently still out of scope (cf. [19]). But reaching beyond semantics to pragmatics is not; neither scientifically nor practically.

References

[1] J.M. Akkermans and J. Gordijn: *Ontology Engineering, Scientific Method, and the Research Agenda*, in S. Staab and V. Svatek (Eds.): Managing Knowledge in a World of Networks (Proc. 15th Int. Conf. on Knowledge Engineering and Knowledge Management EKAW-2006), pages 112-125, Lecture Notes in Computer Science LNCS-4248, Springer-Verlag, Berlin, D, 2006.

[2] G. Schreiber: *Knowledge Engineering*, in F. van Harmelen, V. Lifschitz, B. Porter (Eds.): *Handbook of Knowledge Representation*, Ch. 25, pages 929-946, Elsevier, Amsterdam, NL, 2007.

[3] T. Gruber: A translation approach to portable ontology specifications, *Knowledge Acquisition* **5** (1993), 199-220.

[4] P. Borst, H. Akkermans, and J. Top: Engineering ontologies, *Int. J. Human-Computer Studies* **46** (1997), 365-406.

[5] P. Mika and J.M. Akkermans, Towards a New Synthesis of Ontology Technology and Knowledge Management, *Knowledge Engineering Review* **19**:4 (2004), 317-345

[6] See e.g. the Special Issue on Contexts and Ontologies, *Knowledge Engineering Review* **23**:1 (2008).

[7] P. Mika: Ontologies are us: A unified model of social networks and semantics, *J. Web Semantics* **5** (2007) 5-15.

[8] C. Bianchi and N. Vassallo: Meaning, Contexts, and Justification, in B. Kokinov et al. (Eds.): CONTEXT 2007, pp. 69-81, LNAI 4635, Springer-Verlag, Berlin, D, 2007.

[9] A. Zimmerman, A. Lorenz, and R. Oppermann: An operational definition of context, in B. Kokinov et al. (Eds.): CONTEXT 2007, pp. 558-571, LNAI 4635, Springer-Verlag, Berlin, D, 2007.

[10] F. van Harmelen: *Semantic Web Research anno 2006: main streams, popular fallacies, current status and future challenges*, in M. Klusch, M. Rovatsos, and T. Payne (Eds.), Proc. 10th Int. Workshop on Cooperative Information Agents (CIA-2006), pages 1-7, LNAI 4149, Springer-Verlag, Berlin, D, 2006.

[11] For publications and tools on the e^3 family of networked business ontologies, see www.e3value.com.

[12] T.J.M. Bench-Capon and P.E. Dunne: Argumentation in artificial intelligence, *Artificial Intelligence* **171** (2007), 619-641.

[13] T. Groza et al.: SALT: Weaving the claim web, in K. Aberer et al., Proc. ISWC/ASWC 2007, LNCS 4825, pp. 197-210, Springer-Verlag, Berlin, D, 2007.

[14] I. Rahwan, F. Zablith, and C. Reed: Laying the foundations for a World Wide Argument Web, *Artificial Intelligence* **171** (2007), 897-921.

[15] K. Aberer et al.: *Emergent semantics principles and issues*, in: Database Systems for Advanced Applications Ninth International Conference, LNCS 2973, pp. 25–38, Springer-Verlag, Berlin, D, 2004.

[16] J. Searle: *Rationality in action*, MIT Press, Cambridge, MA, 2001.

[17] J. Habermas: *On the pragmatics of communication*, MIT Press, Cambridge, MA, 1998.

[18] J. Spohrer and D. Riecke (Eds.): Special Issue on Service Science, *Communications of the ACM* **49**:7 (2006), 30-87.

[19] M. Reinders, E. Vinkhuyzen, A. Voss, J.M. Akkermans, J.R. Balder, B. Bartsch-Spörl, B. Bredeweg, U. Drouven, F. van Harmelen, W. Karbach, Z. Karssen, A.Th. Schreiber, and B.J. Wielinga: A Conceptual Modelling Framework for Knowledge-Level Reflection, *AI Communications* **4** (1991), 74-87.

Formal Ontologies Meet Industry
S. Borgo and L. Lesmo (Eds.)
IOS Press, 2008

Knowledge Based Systems in Industry - Ontology Pays Half the Rent

Wernher BEHRENDT, Salzburg Research, Austria

Abstract. We argue that unless the use of formal ontology becomes part and parcel of Systems Engineering culture, industrial use of knowledge based systems will remain a fringe activity and consequently, success stories will remain scarce. The way towards reaping the benefits of formal ontology is to view it as the most powerful instrument for sustainable standardization and as a necessary but not sufficient, condition for building knowledge based systems. Foundational ontology may become the *conceptual* backbone of all interoperable systems, but it will only realize its potential when it is backed by ontology *engines* that compute ontological meaning from ontological descriptions. The paper charts the way from organizational practice expressed in ontologies, to the engineering discipline of building knowledge based systems and it illustrates the difficulties by some lessons learned in a semi-industrial research context in the field of mechatronics.

Introduction - Ontology, Knowledge, Productivity and Marketing

"Ontology" is not part of the every-day vocabulary - neither in business, nor in the news, nor at school, nor in our private lives. "Knowledge" on the other hand, is very much part of the every-day vocabulary. Knowledge is associated with "personal wisdom", "necessity in business", "competitive edge" and "socio-economic success". The "Knowledge Society" and the "Knowledge Economy" have been terms in social studies and in European politics for many years [1], summarizing the above notions. In this paper, we will explore the relationship between ontology and knowledge from a pragmatic point of view: if knowledge is considered an important aspect of our lives and if ontology has something to do with knowledge then it would be prudent to have a closer look at that thing called "ontology". Since we are addressing an audience from industry we will focus on what ontology has to do with organizational knowledge, with industrial processes and artifacts, and with future obligations of enterprises to cater for ever more variety at ever better quality and ever tighter productivity constraints.

Doug Engelbart has been one of the most far-sighted thinkers and doers and his concern has been the "knowledge worker in organizations" [2]. Two of his many ideas are central to the picture painted here: firstly, the organization as a knowledge-processing system, and secondly the taxonomy of "A-B-C" people, where A-people are those who carry out some work and use tools to enhance their productivity. "B-people" are those who build the productivity tools for the A-people. "C-people" are those who build the tools for the B-people. Engelbart argues that the most radical innovations and productivity enhancements occur when C-people change the tools of the B-people.

Aside from knowledge and productivity, there is a third element that has impact on science and technology: marketing. Science has had a long tradition of bad marketing

and what we find today is scientists discovering marketing as a means of attracting funding for their work - with some good results, some bad results and some outright ridiculous ones, in terms of marketing claims and scientific fact. For the purpose of getting the message across, we will be using O'Reilly's marketing notion of "Web 2.0" [3] and Spivak's extension of the idea to "Web 3.0" [4] in order to explain some aspects of ontology in industry. The term "2.0" refers to the emergence of mass phenomena in the WWW and the term of "3.0" refers to the postulated increased use of intelligent or "semantic" systems to support the individual as well as the community when they interact in virtual spaces. However, we should not try and fit everything into the "Web 1.0-2.0-3.0" pattern: over the past twenty years, there have been many more trends at work and the result is much more diverse than can be expressed by O'Reilly's and Spivak's one-two-three formula. What remains useful is the contrast and the attempted confluences of "traditional information systems"; "mass phenomena on the WWW" and "semantic models". The following sections look from these three angles, at organizations, standardization, and software engineering methodology. We then bring back the term "knowledge based systems" to point at the need for computational *machinery* in order for ontologies to become means of enhancing productivity and we illustrate the issues by examples taken from a current project aimed at supporting mechatronic design. We conclude that for enhanced productivity ontology is necessary, but not sufficient and that research in Semantic Web needs to push harder towards integration into systems engineering.

"Organization 3.0" - Ontology as Organizational Practice

Up until the 1980s most organizations had no ontology problem [5], because the business processes happened in the real physical world and were carried out by actors who had learned their trade - this was "Organization 1.0": business processes run by humans and communication problems caused and solved by humans. However, some larger organizations with multiple distributed databases, complex products such as aircraft or computers began to realize that maintenance and synchronization of their systems in response to changing business contexts was beginning to get complicated and expensive, in terms of programming them.

In the 1990s with mergers and acquisitions in business, and with the advent of the WWW interoperation and integration of "legacy systems" became an issue and so, the first wave of knowledge models arrived, partly fuelled by the US research program I3 - intelligent information integration [6]. This was the time of "Ontolingua", KQML - Knowledge and Query Manipulation Language and KIF - Knowledge Interchange Format. What must be said though is that in the 1990s academia and business went rather different ways: the business method of data integration was not based on ontology [7]. Information systems were either ported through bespoke software mappings into the data models used by the new owners after a merger, or relevant data were shipped into data warehouses, transformed and fetched again, by the participant systems. The early 1990s were a time when several technologies arrived at almost the same time: widely distributed systems based on sockets and RPC, the WWW, CORBA, object-oriented programming with C++ and later Java. The remnants of the AI-schools were inventing Description Logic as an attempt to combine logic-based systems with

databases. In the area of business-to-business data interchange, it was the time of STEP [8] for the exchange of product data between manufacturers and suppliers.

The example from applied research is engineering design for mechatronic products [9]. Such goods can range from real-time networked coffee machines in restaurants to aircraft wings or security gates in airports. Quite often, firms with complementary engineering skills team up with solution providers e.g. to offer a complete facilities management solution including sophisticated, software-managed air conditioning and heating as part of the deal. The firms may be from different countries and so they need to set up a virtual organization. For small-to-medium sized enterprises, a large scale schema mapping of their corporate databases would be beyond their means and they would not even want such tight integration for just a few points of collaboration. So they need the ability to very quickly "merge" just their relevant engineering processes and the software support for these. In the research project, we introduced the notion of reference ontologies that cover the domains of mechanical, electronic, and software engineering and we use a generalized engineering life-cycle model (the V-model, in fact) as a prototypical representation of engineering processes. Different engineering tools are connected to the main collaboration server via small adapters for the translation from different CAD/CAE formats into the common reference ontologies. There is a further research assumption, namely that the reference ontologies should be customizable during a set-up phase to form a dedicated "collaboration ontology" which should be regarded as some sort of name space or specific universe of discourse within the full world of mechatronic knowledge.

"Standardization 3.0" - Ontology as Conceptual Industrialization

"Standards ensure desirable characteristics of products and services such as quality, environmental friendliness, safety, reliability, efficiency and interchangeability - and at an economical cost." (http://www.iso.org/iso/about/discover-iso_why-standards-matter.htm). For software-based products, quality, reliability and interchangeability are of particular interest, because these characteristics apply to virtual as well as physical goods. Standards development has always been in response to industrial and societal needs so there is no top-down model for standardization, but increasingly, there is a global realization that the standards process itself has an ontological problem: according to official ISO information, there are approximately 16.500 ISO Standards available, and there are in the region of several hundred in the area of computing and software. Many of the standards overlap in scope and are thus, becoming subject to the ontological question: "how should we organize our conceptualization of the world in order to minimize the number of rules or norms (i.e. standards) and to maximize the compliance of goods with the given norms?".

In the applied research example from mechatronics, dozens of standards are ready to meet (and clash, ontologically speaking): the electronic engineers have to meet compliance norms that hold for the functional components of their design. The software engineers have to follow certain rules according to software development standards (some of these could be built into the collaboration ontology) and the target software has to meet the norms that hold for the intended usage of the end product. Some of the engineering tools have elements of the domain standards inherently built

into the component systems, e.g. tools for structured testing of the connections on a printed circuit board, given a certain electronic design. When standards compliance is not an explicit part of the ontological model, but built into the engineering tools as implicit knowledge, then we are only able to make use of standards at this lowest level of compliance. However, there are ways of "semantic lifting": e.g. we are able to translate from IDF (an electronic data exchange format) into the reference ontology as a first step. The added value comes when we can then relate the identified items to the mechanical world because the reference ontologies (electronics, mechanical engineering) are connected with each other by the "semantic bus" of a foundational ontology which in our case was DOLCE [10]. In other words, we bought ontological interoperability by committing to one foundational ontology. This is "standardization 3.0" in our opinion!

"Software Engineering 3.0" - Ontology and Systems Development Methods

As the WWW developed into the platform of choice for business between 1994 and now (2008), many of the interesting technologies of the early 1990s for interoperation and distributed systems were overrun by a need for absolute simplicity in user interaction (the clickable link) and this trend together with the falling prices for Internet access has ultimately, made the "Web 2.0" revolution possible - this is the world of http, XML and php, at the expense of all other distributed systems technologies. A new type of systems engineer emerged: the LAMP-stack builder (Linux, Apache, mySQL, php). Many web-based content and information systems are built in this way, and there is no place for fancy, difficult to use, difficult to integrate other software. The losers of the simplicity battle moved into market niches: OODBMS are back ends of many CAD systems, CORBA is used in stock exchange applications, Logic Programming disappeared almost completely except in places where applications were able to hide the logic engine and show the user only the results, i.e. clickable links. Description Logic - after a ten year hiatus - re-emerged in the Semantic Web, but is still hard to integrate with conventional distributed systems engineering and this increasing dilemma has led to complete re-inventions, such as Semantic Web Services which attempt to do "Web 3.0" as an alternative technology (including methodology) to all others, hoping for a significant push through big stakeholders in the software market.

What are the difficulties for software engineering, with ontologies? The first and simple question is what functional component of any software architecture is covered by the use of an ontology? In traditional systems, database schemas come closest to the notion of an ontology. In object oriented systems, the actual object hierarchy - without the methods - comes close to an ontology. The trouble with ontologies is that for a software engineer, they are "neither here nor there": the software engineer cannot just put ontology concepts in a schema, nor can they be put into any application code. As a result, a usage pattern has emerged where ontologies serve as a conceptual tool in the earlier stages of design and are then substituted by an object-oriented model, powered by the methods of some application code. Information exchange between systems is done via SOAP based services using some XML-Schema as data messaging format. The resulting systems are at best described as "Web-based data plumbing".

In the collaboration project for mechatronic engineering, the same problem arose: the actual implementation of the system uses an object-oriented distributed system with OODBMS functionality as well as networking services and we are spending R&D effort into translating from the ontological model into a semantically equivalent database model including its application code. Most other "semantic systems" projects face this duplication problem. At its heart is an unresolved paradigmatic conflict - ontologies are useful when you use them for making inferences, but using inference-based systems requires other methods of application building than are in use today.

Knowledge Based Systems 3.0 = Ontology + Inference + Web Systems Engineering

When students learned about knowledge based systems in the late 1980s, they were told the basic architecture of such systems: user interaction layer, knowledge base, rule base, inference engine. The knowledge base contained the "static" model whereas the rule base contained the "dynamic model". The inference engine was the basic mechanism by which the rules were applied to the knowledge base. The problem of integrating knowledge based systems development with conventional software engineering was identified then as the major inhibitor. However, while people were trying to work out how you could combine OO languages with AI languages and how you could combine relational databases with description logics, and how you could mix declarative rules and procedural application code, a certain Tim Berners-Lee (TBL) had the idea to access text and data files that were distributed over a network of machines in CERN, and to make them accessible via a simple protocol and to format them via a simple formatting language. "Web 3.0" is the attempt of computer science to recover from the impact of "Web 1.0". TBL's idea of the Semantic Web [11] is his attempt to assist this recovery. What has been underestimated is that we still need a systems engineering approach that can reconcile the programming paradigms, as we did before. The complexity of this task is in fact, caused by the Web's astonishing simplicity of use.

References

[1] P. Drucker, *The Age of Discontinuity*. London: Heinemann,1969

[2] D. C. Engelbart, Toward High-Performance Organizations: A Strategic Role for Groupware. *Proceedings of the GroupWare '92 Conference*, San Jose, CA, August 3-5, 1992, Morgan Kaufmann

[3] T. O'Reilly, What Is Web 2.0 ? Design Patterns and Business Models for the Next Generation of Software, http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html

[4] N. Spivack, The Ontology Integration Problem, personal weblog from August 2006, http://novaspivack.typepad.com/nova_spivacks_weblog/2006/08/the_ontology_in.html

[5] M.S. Fox, M. Barbuceanu, M. Gruninger, "An organisation ontology for enterprise modelling: preliminary concepts for linking structure and behaviour," wetice, p. 71, *4th Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET-ICE'95)*, 1995

[6] G. Wiederhold, Intelligent Integration of Information, *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp 434-437, May 1993.

[7] M. L. Brodie, M. Stonebraker, *Migrating Legacy Systems*, Morgan Kaufmann Publishers, 1995.

[8] STEP Application Handbook ISO 10303 VERSION 3, 2006

[9] V. Damjanovic, W. Behrendt, M. Plössnig, M. Holzapfel, Developing Ontologies for Collaborative Engineering in Mechatronics, *Proceedings of the 4th European Semantic Web Conference (ESWC 2007)*, Innsbruck, Austria, June 03-07, Springer LNCS, 2007.

[10] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, A. Oltramari, A. WonderWeb D18, Ontology Library (final), 2003.

[11] T. Berners-Lee, J. Hendler, O. Lassila, The Semantic Web – A New Form of Web Content That is Meaningful to Computers will Unleash a Revolution of New Possibilities, Scientific American, May, 2001, http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21

Formal Ontologies Meet Industry
S. Borgo and L. Lesmo (Eds.)
IOS Press, 2008

Utilizing Ontologies for Petrochemical Applications

Eric LITTLE[a1], Joseph EBERLE[b] and Fred TURINO[b]
[a] *Director, Center for Ontology & Interdisciplinary Studies, D'Youville College, Buffalo, NY, USA*
[b] *Computer Task Group (CTG), Buffalo, NY, USA*

Abstract. This paper discusses some of the current challenges for knowledge exploitation and sharing in petrochemical domains and provides one approach to utilizing ontologies to solve some of those problems. By building ontological profiles that accurately classify kinds of crude oils, petroleum products and refineries, the ontology proved capable of providing significant inferential capabilities that were useful for optimizing numerous phases of the petrochemical life cycle.

Keywords. Ontology, petrochemicals, business intelligence, knowledge sharing, ontology profiles, OWL

Introduction

This paper represents work performed for a large multinational petrochemical company over a three month period to produce a prototype ontology which could later be expanded to a production level system. These kinds of companies face extreme challenges in efficiently capturing, transferring, and taking action on various kinds of information integral to their business success [1]. Due to a combination of the size and corporate structure of most petrochemical companies, it is increasingly difficult to link segments of the corporation together globally, even though they are often working with overlapping data sets [2]. Areas such as exploration, production, refining, marketing, and trading often operate in relative autonomy from one another. This results in problems of knowledge sharing, where, often times, information gleaned by one group of employees is rarely transmitted to, or utilized by, others within the same corporation. While this phenomenon is not specific to petrochemical corporations (i.e., one can find this in many corporate settings such as pharmaceuticals, manufacturing, national defense, and elsewhere), our discussion will focus specifically on the petrochemical industry and provide a means for utilizing ontologies to overcome some of the pitfalls faced therein. Our approach showed a capability to better integrate various segments of the petrochemical corporation under consideration and provide a means to increase their *business intelligence* [3, 4], which can be understood generally as a set of concepts and methods used to improve business decision-making through the use of

[1] Corresponding Author: Eric Little, Director, Center for Ontology & Interdisciplinary Studies, D'Youville College, 201 Porter Ave, Buffalo, NY,14201, USA; email: little@dyc.edu (or) eglittle@hotmail.com; fax: 716-692-0313

fact-based tools, such as ontologies, where one is capable of capturing, modeling and utilizing pertinent categorical and instance-level data. While there are several potential methodologies currently available for application in this area [5-8], we have incorporated an approach following from [9-12].

1. Some Current Problems of Knowledge Capture, Exploitation and Sharing in Petrochemical Industries

The domain of petrochemicals is highly complex. For this reason, the petrochemical industry contains many types of highly educated individuals who gather and utilize various kinds of information about different segments of the overall process. Examples of these segments of the petrochemical process include the gathering and utilization of: geological data, geographical data, basic and advanced chemistry, refining processes, marketing techniques, and commodity trading, to name but a few. Due to the nature of these distinct, but overlapping, facets of the overarching processes associated with petrochemical discovery, production and transformation, the industry faces many unique challenges regarding knowledge capture, knowledge transfer and ultimately the ability to take action based upon vast amounts of data and a wealth of tacit corporate knowledge. The following represents a tertiary list of considerations and challenges in the petrochemical domain:

- **Data Overload** – Where large amounts of data specific to a single activity or result fails to be properly captured, filtered or transferred to other departments. For example, exploration data contain essential facets of business knowledge that is critical to the entire petroleum processing lifecycle yet this essential data often gets lost because it is mixed in with an array of assay data (i.e., chemical information pertinent to identifying – or fingerprinting – a crude type) that is far too large and complex to be easily consumable. The initial data generated from seismology and early exploration produces hundreds, if not thousands, of data points. A specific example is within a typical exploration assay, where over 275 points of data are gathered and passed downstream, however, only about 8 of those points (Sulfur, Density, Acidity, and Cut Points) are critical to the downstream consumer.
- **Departmental Focus** – Refinement teams learn essential information that is valuable to both upstream logistics and exploration and downstream distribution and marketing. Yet because refinement engineers, for example, are chartered only with improving the refining processes, there is little to no incentive to examine other types of related data and assimilate/coordinate their activities with other departments.
- **Geographic and Cultural Diversity** – By nature, large petrochemical organizations are divided by geographic and cultural boundaries that effect the flow of best practices and knowledge transfer.
- **Multi-Perspectivalism** – Data and knowledge is abundant within the petrochemical industry yet, because of varied perspectives of distinct departments, much of this knowledge gets lost. For example, regarding one single piece of data (density), the financial department is concerned with basic crude categories of Light or Heavy, as they pertain to price per barrel and the

futures markets. The exploration team is concerned with flow characteristics related to the density in order to identify how best to extract the crude. Refinery engineers are concerned with density in terms of how it defines what blending and processing techniques are required to produce a given petroleum product. Because each group looks at essentially the same data, the failure to utilize a common semantic structure that can support varied levels of granularity inhibits overall knowledge transfer and the ability to re-use and repurpose information for a multitude of perspectives (see Figure 1).

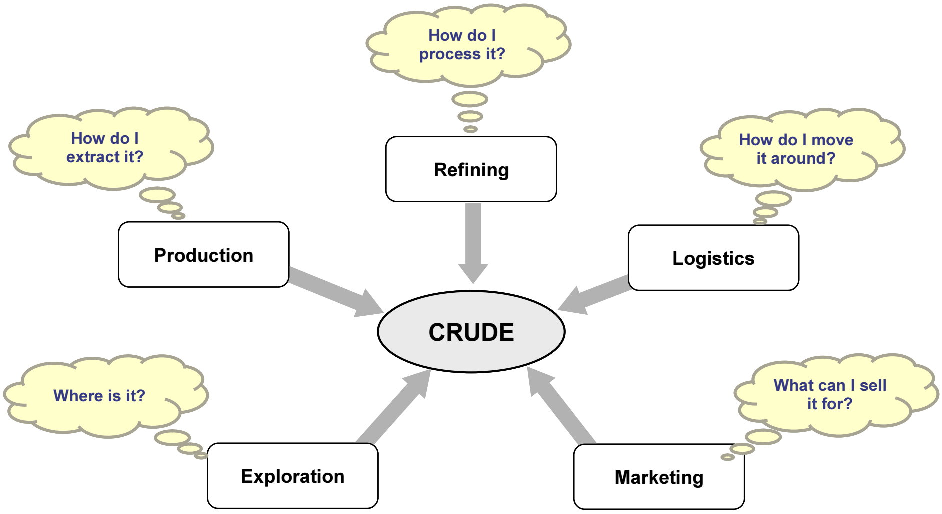

Figure 1: Multi-Perspectivalism in the Crude Industry.

1.1. Utilizing Ontologies to Establish a Culture of Knowledge Sharing

In order to establish an enterprise-wide knowledge domain, the ontology must provide an incentive for knowledge sharing and collaboration by establishing direct and apparent benefits to the various contributing departments and experts within the organization. By providing direct results and benefits to its contributors, the knowledge domain can incrementally grow based upon the availability of new forms of knowledge that are highly relevant and beneficial to those using it.

While producing the ontology, the process itself fostered a great deal of intellectual stimulation on the part of various subject matter experts (SME's) and, in turn, promoted a desire on their part for increased information capture and distribution. Petrochemical engineers, seismologists, geochemists, and other researchers within the organization quickly discovered many semantic and logical issues within the petrochemical domain, which, beforehand, they had not considered. For example, initial research into existing assay data pointed out the need for better understanding the necessary conceptual distinctions between substances (e.g., sulfur, nickel, vanadium, etc.) and their corresponding attributes (e.g., flow characteristics, acidity, density, etc.). These items appear together on assay tables and have been traditionally conflated with one another. Within the ontology's structure, however, it was immediately apparent to the members of the organization that in order to provide appropriate reasoning

capabilities within the system, defining the ontological distinctions between basic categorical items such as substances, attributes, or between spatial and temporal items was paramount to the development of a successful operational system. In this sense, the ontology performed the service of providing a better formal framework of items found within the petrochemical domain, which previously was implicitly understood by members of the organization, but not explicitly spelled out in a cogent and coherent manner (see Figure 2).

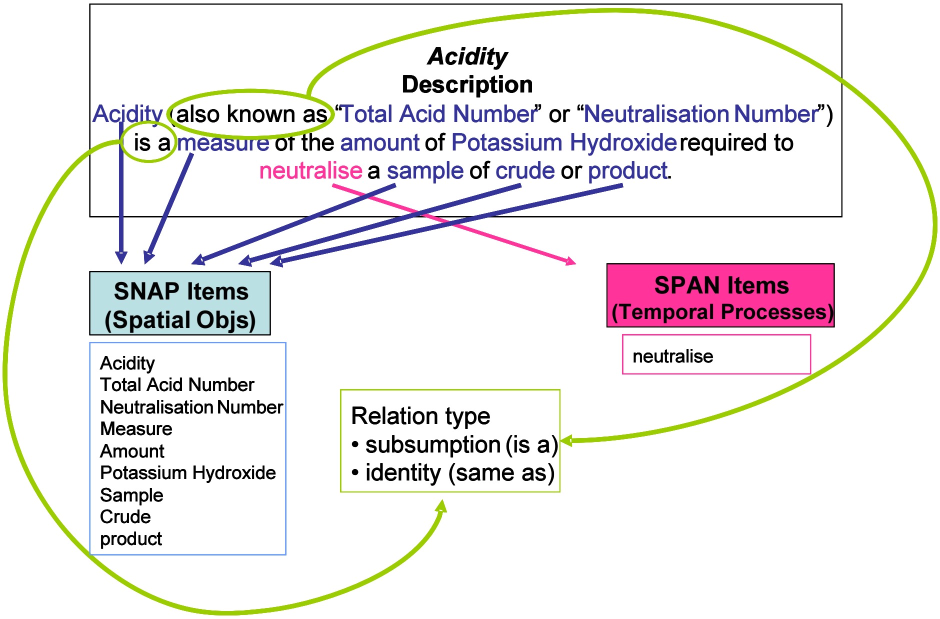

Figure 2: Conceptual Decomposition of Acidity Definition.

In order to capture these kinds of distinctions, SME participation in the ontology design process, particularly at the domain-level of consideration, is crucial [9]. This means that formal ontologists should provide a means for SME's to better understand their domain in a formal way, while at the same time, taking care to not transform or alter that domain information. A transformation of the domain-specific data could result in negative consequences, such as the case where an ontology is designed which does not accurately reflect a given domain of interest or the corresponding knowledge of SME's about it. In this sense, ontologists must be able to formulate a rather exhaustive understanding of the particular domain (or sub-domain) in question with significant help from domain SME's. Our ontology staff, for example, moved from little or no knowledge of the domain, to having a great deal of knowledge about the entire end-to-end process of the petrochemical process. The domain knowledge was gained through two general processes: 1) independent research of petrochemical documentation (e.g., published reports/documents/lexicons, web information, industry standards/protocols) and 2) close collaboration with domain SME's (e.g., interviews, weekly meetings, generation and review of progress reports and segments of the ontology).

When placed in the hands of petrochemical subject matter experts, the ontology should provide a clear and understandable formalization of the petrochemical domain

that the SME's will take pride in and ownership of, since their domain knowledge is integral to its construction. We found our SME's to be enthusiastic about the ontology and with the positive results it provided. They began to immediately see the direct benefits of being able to integrate the expertise of SME's in related subject areas to produce an overarching knowledge repository.

1.2. Challenges to Constructing a Common Lexicon and Ontology Model

While performing domain analysis, and prior to establishing an ontological model, one of our first exercises was to establish a common petrochemical lexicon through basic domain research and with the help of the organization's SME's. The lexicon contained several hundred technical and industry-specific terms, which provided an initial framework of the domain's semantics. It was constructed by utilizing the following:

- on-line (open source) glossaries of general petrochemical terms [13]
- corporate literature and presentations published by domain SME's
- direct collaboration and discussion with domain SME's

As pointed out in [9, 12], the lexicon should remain an open research item, since new terms may need to be constantly inserted into the ontology based on new areas of interest, changes in the understanding of the domain, or the creation of new technologies. It is often essential to provide a mechanism (e.g., wiki, website, or other form of collaboration), where individuals can update the ontology's lexicon. Care must be taken to provide documented annotations for any and all changes to the lexicon (or the ontology itself). Additional research and analysis is often needed in cases where domain information is sparse.

One major challenge to this phase is generally referred to as the *knowledge engineering paradox*, which states that the more competent a SME becomes within their domain, the less able they are to accurately and formally describe the knowledge they utilize for various problem-solving techniques [14, 15]. Often, when SME's discuss domain-specific information, functions and processes they bring with them certain pre-built assumptions that are not explicitly available to those with whom they are communicating. To combat this paradox, we interrogated subject matter experts from various departments to provide further clarification. We utilized the formal structure and inherent relationships within their domain descriptions to formally define all relevant SME knowledge. We then validated the annotated ontology model with the SME's themselves, to make sure we accurately captured the domain knowledge accurately. When conflicting opinions arose, we forced them to either 1) arrive at a common consensus on the issue/terminology, or 2) formally capture the distinctions through the construction of new, more constrained terminologies. This exercise provided the following critical results:

- A normalized domain terminology
- An improved understanding of the hierarchy associated with those terms
- A greater understanding of disparate petrochemical data between various domain SME's
- An ability to establish and reinforce the project scope and direction

2. General Methodology for Designing Upper-, Mid- and Domain-Level Segments of the Ontology

We incorporated the BFO upper-ontology schema [16-18] for our petrochemical ontology, since it provided a sufficiently broad and flexible basis upon which to capture the kinds of items found in the petrochemical domain. It should be noted that other upper ontologies such as DOLCE and SUMO are also available for such applications, but each of these upper ontologies provides a slightly different metaphysical approach and accompanying formal machinery [19-21]. We chose BFO because we sought an upper ontology that was flexible, realist-based and not overly committed to set theoretic formalisms (i.e., one which also provided a mereotopological approach). Categories were bifurcated into their respective spatial (i.e., SNAP) and temporal (i.e., SPAN) components and subsequently related via their corresponding trans-ontological relations. A small depiction of the upper-to-mid-level ontology can be seen in Figure 3.

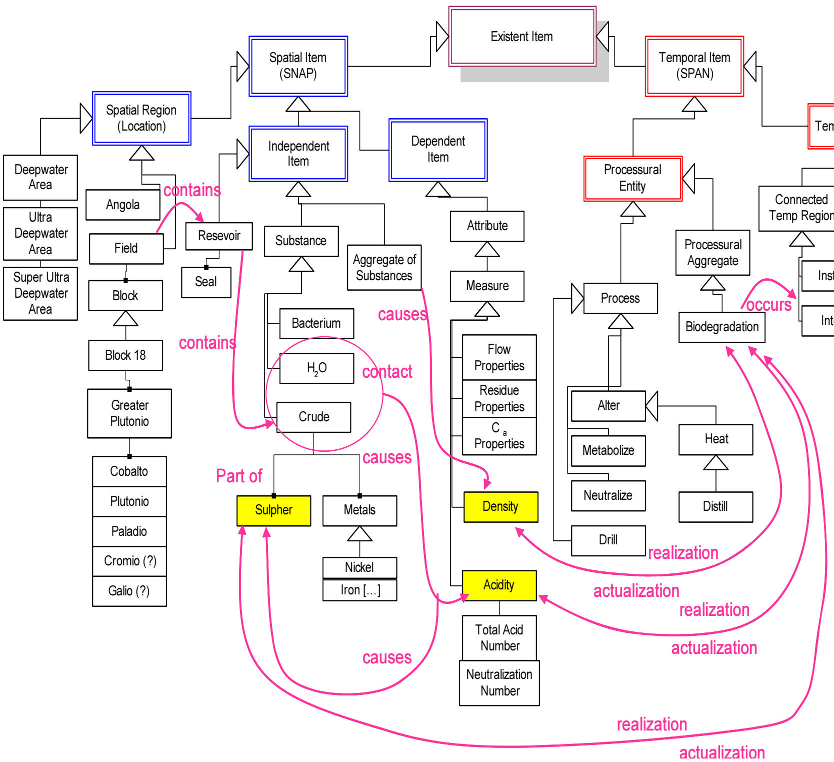

Figure 3: Basic Upper-to-Mid-Level Ontological Categories for Acidity, Sulfur and Density

The items to the left represent SNAP elements of the ontology, whereas items to the right represent SPAN. The open-headed arrows show subsumption relations found within the basic categorical hierarchy of the ontology (i.e., is-a relations). Other connectors show more complex relations such as part-hood (e.g., between Crude and Sulphur/Metals) or in the case of the curved arrows, can even show *transcategorical* relations such as containment or causation, where relations can exist between different SNAP items (e.g., substances and properties), between SPAN items (processes

occurring over intervals of time), or between SNAP and SPAN (realization and actualization). All complex relations in Fig. 3 (those not captured specifically in the hierarchy) were captured as OWL Properties of those classes.

The ontology was constructed in Protégé (OWL-DL), with RacerPro as the system's reasoner. 278 RDF triples were initially constructed to account for the basic facts of the portion of the petrochemical domain under investigation (the BFO upper level plus mid-level categories and relations). Utilizing the RacerPro reasoner allowed the system to infer an additional 450+ RDF triples.

The scope of the ontology was constrained so as to focus on the relationships between three basic categories associated with crude oil and its associated petrochemical products – namely, *sulfur, acidity* and *density*. A full-scale petrochemical ontology for industry utilization would require the categorization of more than 250 such entities, corresponding to all columns found in petroleum assay tables. Sulfur, acidity and density were chosen due to the fact that those three elements of a crude oil can be utilized by domain SME's to ascertain a large amount of information about a given crude commodity.

Density can be understood as the specific gravity of a given crude and is widely used in the petrochemical industry because products are bought, sold, stored and shipped in volumetric and weighted units of measure. Density readings correspond in the ontology to the general categories of *light crude* (density/specific gravity < 40) or *heavy crude* (density/specific gravity of 20 or less). It should also be noted that wax content can also affect the lightness or heaviness of crude as well, though this facet was not captured in the ontology being described. Lighter crude is easier and cheaper to refine than heavier crude, making it a more valuable commodity for the production of petroleum products such as gasoline, diesel, or aviation fuel [22].

Sulfur content in crude corresponds to the ontological categories of *sweet crude* (sulfur content < 0.5%) and *sour crude* (sulfur content > 0.5%). Like density, sulfur content dramatically affects the refining process and associated costs, since crude oil with a higher sulfur concentration must undergo more refining to remove it from the petroleum products produced.

Acidity is another highly important aspect of a crude oil commodity, since highly acidic crude (a total acid number (TAN) > 0.5) corrodes pipelines in refineries much more quickly, affecting performance of the machinery and ultimately causing more maintenance issues. The ontology contains categories for *highTAN crude* (TAN > 0.5) as well as *lowTAN crude* (TAN < 0.5).

3. Utilizing the Ontology to Create Categorical Profiles

The general classifications or profiles of crude oils, refineries, petrochemical products and the like are well established and widely known within the petroleum industry. The variables that differentiate between light-sweet and heavy-sour crude can be established within the ontology allowing for crude types that can be easily classified for general commodities markets. However, in order to better understand and share basic, common-sense information about crude composition, the ontology proved useful for producing ever more detailed profiles. For example, crude profiles based upon specified cut points and impurities can be used by the system to reason about how general crude profiles match to specific petroleum products or specific refinement processes. Therefore, it is possible in the ontology constructed to classify crude as

"Good Jet Fuel" and take the action of sending that crude to a refinery specifically optimized to produce jet fuel (see Figure 4).

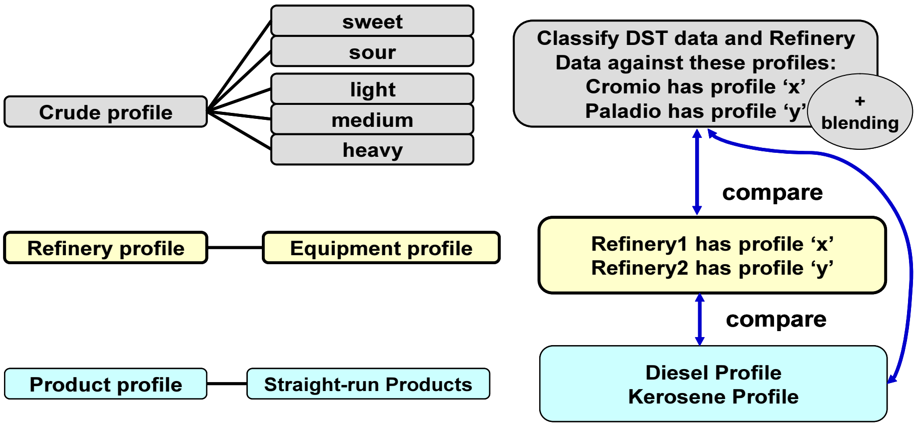

Figure 4: Profiles Based on Ontological Categorization.

3.1. Commodity Crude Classification

From the basic categories contained in the upper- and mid-level ontology levels, along with their temperature-based cut points (i.e., fractions based on temperatures to which the crude is heated for processing), the system was able to ingest raw assay data (stored in databases) and correctly determine combinatorial categories such as *light sweet highTAN*, *heavy sour highTAN, light sour lowTAN*, etc. This allowed the system to rapidly identify specific categorical commodities of interest to refining engineers or commodities brokers to support activities such as buying and selling products, determining the types of crude being produced from certain geographic regions, etc.

3.2. Product Classification

Geographic and seasonal demands define a great deal of what optimizes petrochemical production. By establishing product profiles we were able to classify crude types that where particularly well-suited for specific uses and products. Product profiles can, in turn, be used to adjust the capacity of a refinery and/or redirect crude shipments based upon seasonal and regional product demand. For example, Southeast Asia has a particularly high demand for Kerosene, while Southern California has a high demand for low-emission automobile fuel (gasoline). By establishing product profiles, we were able to demonstrate the ontology's ability to quickly adjust the Jet Fuel profile to seasonal variations as other situations such as when the weather gets much colder in certain geographic regions.

3.3. Refinery Type Classification

Once crude types could be profiled based on raw assay data, a further goal of the ontology tool was to establish ontological links between upstream production data and downstream processing needs. We constructed refinery profiles in the ontology that represented its appetite for specific crude types. SME's helped to identify the different

classes of refineries based on the specific types of refining equipment that they contained (light ends units, cokers, etc.) and which are needed to perform certain chemical transformations or processes. We then studied each of the companies 18 refinery schemas to determine which major refining equipment was present. This is important because a refinery without a coker is ineffective at processing heavy crude; a refinery without desulphurization units cannot effectively handle sour crude (i.e., highly sulfuric crude).

To accomplish this task, we but several categorical profiles of refineries, based on their components. Thus, refineries that contained coking capabilities were labeled as *coker refineries*, and rules were written that allowed the system to infer that coker refineries are best equipped to handle heavier crude oils, whereas a refinery with a light ends unit (LEU) is particularly effective in dealing with lighter crude types which possess large top cuts (e.g., crude oils that contain significant amounts of gases such as propane). By defining the major equipment contained in each refinery, the system was able to effectively aid in logistics, prediction of refining costs, issues of current capacity, blending recommendations, and other relevant decision-making processes. Using this technique, the system was able to automatically assign specific logistic actions such as “a particular well is currently producing a *heavy sour crude*, which should be sent to a specific set of refineries that contain both a coking capability as well as a desulphurization unit.”

3.4. Refinery Crude Classification

A more detailed classification was then established by determining a specific refinery’s throughput capabilities, which, in turn, can be used to establish a refinery’s optimal crude input requirements. We therefore, built a more detailed classification in our ontology in order to establish refinery profiles that represented the refinery’s appetite for specific types of crude. To do this, we worked with SME’s to examine the schematics, specified cut-points and throughput statistics of the refineries based upon crude types. We then established an optimal crude profile for each of the 18 refineries under investigation, which would serve to optimize the overall flow and throughput of the distillation towers and the various processing units.

Using this detailed classification we where able to assign specific inferential capabilities within the system, such as identifying the optimal refinery to handle the crude. We were also able to take further action by defining blending instructions of various crude types that would establish the optimal blend needed to meet the refineries appetite or demand on the part of a specific geographic region.

4. Conclusions and Further Research

The ontology constructed provided a pilot-level, small-scale ontology, which offered scientists and engineers a means to infer across numerous sub-domains of the petrochemical industry (from exploration through production to refining) producing positive results that, previous to its instantiation, were not possible. Petrochemical engineers could utilize the system to exploit both their large data sets, as well as infer new kinds of relations when data was sparse. A larger more robust ontology, designed with the same characteristics, would provide even more categorical fidelity and improved reasoning capabilities. For example, by establishing deviation profiles in our

ontology, we where able to establish a means of automatic inference of price per barrel of new crude assays. This extended capability could be used to support automated bidding and generate sell offers for commodities markets.

One area of research that is still ongoing is to better utilize OWL-DL ontology tools to more effectively handle the kinds of advanced mathematics prevalent in many of the complex problems in the petrochemical domain. Protégé with RacerPro provided little to no capacity to handle sophisticated math formulas or provide inference on much more than rudimentary cardinality. Since this project, we have incorporated more sophisticated OWL-DL tools such as the Top Braid Composer and Ensemble products from Top Quadrant, which provide superior modeling and reasoning capabilities than Protégé. It is important to understand the role of ontologies in complex industries such as the petrochemical domain, since it is conceivable that large-scale ontologies may even extend beyond the capabilities of OWL-DL, requiring tools that employ Common Logic or F-Logic. The petrochemical industry has spent a great deal of time effort and money to develop high-end software in areas such as geographic mapping, analysis, and exploration. It is highly unlikely that ontology-based tools (especially those built on RDF) are going to provide greater insight to these highly math-intensive, task specific engineering tools and models. Nonetheless, this study showed that, in a more global sense, ontologies offer the ability to provide an overarching business intelligence that supports and extends many of these pre-existing tools, by providing the semantic underpinning that can better integrate these other math-based tools. In this sense, existing tacit knowledge can be effectively leveraged and related for improved reasoning capabilities across disparate regions of the petrochemical domain.

References

[1] P. Leavitt. Applying Knowledge Management to Oil and Gas Industry Challenges, American Productivity and & Quality Center, Oct., 2002.

[2] M. C. Jones & R. L. Price. Organizational Knowledge Sharing in ERP Implementation: Lessons from Industry, in M. A. Mahmood (ed.) *Advanced Topics in End User Computing*, Harrisburg, PA: Idea Group Inc. (IGI), 2002.

[3] S. Williams & N. Williams. *The Profit Impact of Business Intelligence*, Elsevier Inc., 2007.

[4] E. Vitt, M. Luckevich & S. Misner. *Business Intelligence*, Microsoft Corp., 2002.

[5] A. Gómez-Pérez, J. Ramírez, B. Villazón-Terrazas. Methodology for Reusing Human Resources Management Standards. SEKE 2007: 280-285, 2007.

[6] A. Gómez-Pérez, A. López-Cima, M. C. Suárez-Figueroa, Ó. Corcho. The ODESeW Platform as a Tool for Managing EU Projects: The Knowledge Web Case Study. EKAW 2006: 389-396, 2006.

[7] M. Cristani & R. Cuel, Ontology Methodologies: a survey. Invited paper on *International Journal on Semantic Web and Information Systems*, Jan-March, 2005.

[8] L. Obrst, S. Semy, M. Pulvermacher. Upper Ontology Distinctions for Use in U.S. Government and Military Domains. Submission to the Third International Conference on Formal Ontology in Information Systems, FOIS-04, Torino, Italy, November 4-6, 2004.

[9] E. Little. A Proposed Methodology for Application-Based Formal Ontologies, *Proceedings of the Workshop on Reference Ontologies vs. Application Ontologies*, 15-18 Sept., University of Hamburg, CEUR-WS.org., 2003.

[10] B. Smith. Against Idiosyncrasy in Ontology Development, in B. Bennett and C. Fellbaum (Eds.), *Formal Ontology and Information Systems*, (FOIS 2006), Baltimore November 9-11, 2006.

[11] E. Little. On an Ontological Foundation for Work Domain Analysis, CH 11 in *Applications of Cognitive Work Analysis* (Bisantz and Burns, eds.), Lawrence Erlbaum Press (2008 - in press).

[12] E. Little & L. Vizenor. Principles for the Development of Upper Ontologies in Higher-level Information Fusion Applications, in *Proceedings of the 2006 Formal Ontology for Information Systems Conference*, Baltimore, Maryland, November 9-11, 2006.

[13] The Society of Petrophysicists and Well Log Analysts (SPWLA), http://www.spwla.org
[14] F. Hayes-Roth, D. A. Waterman & D. B. Lenat. *Building Expert Systems*, London: Addison Wesley, 1983.
[15] S. Reisman. *Multimedia Computing: Preparing for the 21st Century*, Harrisburg, PA: Idea Group Inc., 1994.
[16] B. Smith. Basic Formal Ontology, http://ontology.buffalo.edu/bfo/, 2002.
[17] P. Grenon. Spatiotemporality in Basic Formal Ontology: SNAP and SPAN, Upper-Level Ontology and Framework for Formalization, *IFOMIS Technical Report Series*, (http://ifomis.de), 2003.
[18] See BFO Home Page, http://www.ifomis.uni-saarland.de/bfo/
[19] A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, L. Schneider. Sweetening Ontologies with DOLCE, Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web : *Proceedings of 13th International Conference, EKAW* 2002, Siguenza, Spain, October 1-4, in Lecture Notes in Computer Science, Vol. 2473, pp. 166 ff., 2002.
[20] Standard Upper Ontology Working Group (SUO WG) http://suo.ieee.org/SUO/Ontology-refs.html
[21] P. Grenon. BFO in a Nutshell: A Bi-categorial Axiomatization of BFO and Comparison with DOLCE, IFOMIS Report ISSN 1611-4019, 2003.
[22] See http://moneyterms.co.uk/crude-oil/

Formal Ontologies Meet Industry
S. Borgo and L. Lesmo (Eds.)
IOS Press, 2008

Implicit Metadata Generation on the Semantic Desktop Using Task Management as Example

Olaf Grebner[a], Uwe Riss[a]
[a] *SAP Research, Karlsruhe, Germany*
{olaf.grebner, uwe.riss}@sap.com

Abstract. The daily work activities of knowledge workers (KWers) are characterized by a highly dynamic working style that challenges support by task management (TM) systems. In this paper we present the KASIMIR personal task management system that operates on Nepomuk Social Semantic Desktop. KASIMIR features a combination of a task sidebar with a set of task plug-ins into common desktop applications such as email clients and web browsers to tightly integrate the TM into the KWer's knowledge production activities. Thereby, KASIMIR exploits a KWer's task management activities to seamlessly create metadata within the KWer's personal semantic network. Based on this example, we present a mechanism that leverages actions the KWer performs anyway within the knowledge work process to create metadata. This is a powerful alternative to the common problem of enterprise environments that KWers lack the willingness to contribute effort to create annotations without receiving direct benefit. The evaluation of KASIMIR's task plug-in approach showed that it was well received.

Keywords. Task management, Semantic Desktop, Knowledge Management, Personal Information Management.

Introduction

In this paper we show how a task management system can leverage semantic technologies to support Knowledge Workers (KWers) such as researchers and managers more efficiently. These technologies provide a framework to better manage, integrate and find information and allow for inferences based on semantic annotations.

However, when looking at the creation of semantic annotations in an enterprise scenario, today's solutions suffer from the problem that creating semantic annotations is costly. For example, it requires the KWer or a dedicated knowledge manager to invest significant effort in maintenance. Whereas ontologies can be rather efficiently built today, the creation of instances requires a permanent effort. Generally KWers avoid such activities since they do not directly contribute to their work. The extra maintenance effort – and especially the perceived extra work [1] – leads to weak

acceptance among the workforce already being under constant time pressure. Another practiced alternative consists of 'mining' semantic annotations using data and text mining methods (see [2] and references there). However, this alternative still has to prove its applicability in the mainstream enterprise domain.

To address the challenge of metadata creation and maintenance we follow an approach of implicit metadata handling. By this we mean that we want to directly derive metadata from the KWers' work activities. As an ideal leverage for this approach we have identified Task Management (TM), i.e., that part of knowledge work (KW) that essentially consists in organizing the information objects (IOs) on the desktop with respect to activity planning. Since almost all IOs are related to any work activities, TM provides an ideal medium for semantic description of IOs.

However, today we face the basic challenge that current approaches to task management do not provide an effective integration with the work context of the KWers and the tools they employ in their daily work. In particular, existing TM software does not efficiently support information management, e.g., it requires KWers to copy and paste information from different application contexts to their TM tools [3]. This results in duplication of information and work. Moreover, it increases the cognitive load and introduces additional administrative overhead for the KWer and reduces its relevance and value to KWers, whose motivation to use it is clearly reduced. Compared to the required effort these tools do not provide sufficient benefits.

As a consequence TM is not widely enough used to ensure a complete grasp of sufficiently large number of IOs. The KWers' daily work process is dominated by sending and processing emails, web browsing, working on documents and doing calculations in spreadsheets [4]. Thus emails, websites, documents, spreadsheets, etc. represent central, *document-like information objects* which are related to work activities. To use TM as a medium for semantic integration of IOs on the desktop, we first have to ensure that KWers get a TM infrastructure that enables them to easily handle their tasks and the related IOs.

In the current approach we have realized a TM infrastructure with semantic web technologies. Driven by the application domain, the guiding principle has been the idea that human action (as production activity) and knowledge (including semantics) are closely interwoven [5]. To this end the presented TM infrastructure exploits the work activities of a KWer as the primary means to derive semantic annotations. This addresses the above mentioned central challenge of today's semantic web technologies. Integrating semantics into a TM system goes way beyond the usual coupling of email tools and TM systems as part of PIM, e.g., as reported in [6].

In the following, we focus on the aspect of implicit semantic annotations creation out of the work activities of KWers. The remaining paper is structured as follows: First, we present the KASIMIR prototype for work-process embedded TM. Second, we show how this prototype leverages the semantic foundations of the Nepomuk Social Semantic Desktop (SSD). Third, we show how KASIMIR extracts metadata seamlessly from the KWer's task management actions and present the abstracted model of this mechanism. Fourth, we present the results of the prototype evaluation. Finally, we conclude with some ideas for further work.

1. KASIMIR prototype - work process embedded TM

The primary motivation for KWer to use task representations to manage their activities can be seen in the fact that it allows them to efficiently organize and prioritize their work. To realize an efficient support in this respect, we propose with KASIMIR a combination of a set of *task plug-ins for desktop applications* and a TM application in the form of a *task sidebar*. Using the task plug-ins, KASIMIR is a personal task management application that is fully integrated in the KW process.

On the basis of these task plug-ins, the presented KASIMIR application provides a personal task management process support to the KWer which is directly embedded in the applications that the KWers use to perform their KW activities. Through the integration of TM into the applications the KWers regularly work with, the effort of handling personal tasks is reduced to a minimal amount.

With KASIMIR, the KWer is able to manage the personal tasks in a *task sidebar*. This encompasses all activities of the above presented personal task management process. It presents all task-related information, such as e.g. documents, collaborators etc.. It supports the task execution through collecting all task-relevant information in one place. As well, the KWer can organize and prioritize a set of tasks in a task list, which is accompanied to filtering possibilities.

The desktop *application plug-ins* enable the handling of tasks in the context of already existing information objects so that KWers can bring their personal information directly in relation with tasks from within their production knowledge work process. The goal is to support this process across application boundaries. The plug-ins support the personal task management activities as introduced above, e.g., the KWer can collect and organize task-related information. On the one hand, the task plug-ins support fast access to tasks from those applications, which are well known to the KWers, automatically relating the IOs in these applications to the respective tasks. On the other hand, they display task information directly in the KWers' applications and enable task management functionality from there.

Our preliminary user study showed that *email, calendar and internet browser* are the IOs with which KWers work mostly within their knowledge work process. KWers spend as well most of their working time with applications maintaining these IOs [7]. Thus, we implemented task plug-ins for Microsoft Outlook and Mozilla Firefox covering these IOs, see Figure 1.

Using the task plug-ins, the KASIMIR prototype supports the KWers' personal TM activities. While working with an IO during the knowledge work process, the KWers identify that the IO can be associated to an existing or a new task. Using the application task plug-ins, the KWers can now define the relations of IOs to tasks, i.e., the direct association to an existing task or the creation of a new task. For example, a document-like IO such as an email triggers a follow-up task for a KWer. For this, the KWer can create a task directly from the email client. The desktop application plug-ins, e.g., within the email client, then transfer as much information as possible from the IO to the task.

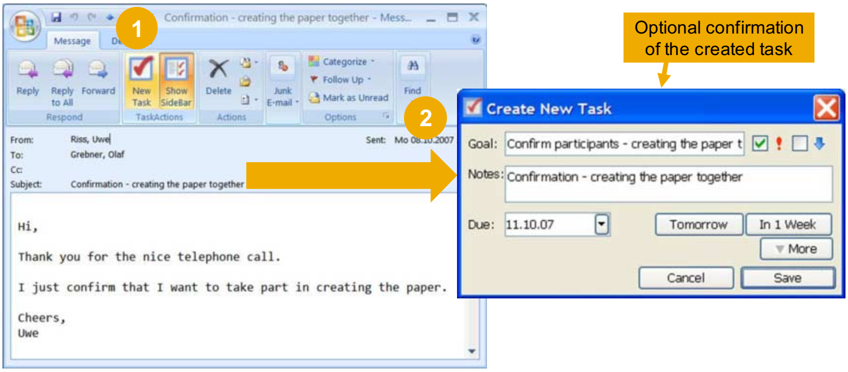

Figure 1. Task creation from an email.

KWers can supplement additional information to make the task more precise. Ideally the task creation works without any additional effort by the KWer. For example, the email plug-in extracts some email information and fills task properties accordingly. In this scenario, the email subject pre-fills the task description and the senders and receivers of the email represent recommendations for task collaborators. We'll explain in section 3 that these senders and receivers are resolved to the corresponding person concepts of the KWer's personal knowledge base, as they are used as structuring IOs. The KWer can additionally provide other information, e.g., a due date, but is not required to do so for creating a task.

The TM sidebar closely interacts with the task plug-ins, i.e., the task plug-ins can control the TM sidebar remotely. For example, when a KWer reads an email related to a task, the TM sidebar shows the details of the particular task.

For Mozilla Firefox, the task plug-in provides the same functionality as presented for the Microsoft Outlook plug-in. The KWer can create a task from or relate a task to a website from within the web browser. Furthermore, the task plug-in presents the tasks related to a website within the website.

2. KASIMIR on the Nepomuk Social Semantic Desktop

The KASIMIR prototype (including task sidebar and plug-ins for Microsoft Outlook and Mozilla Firefox) builds upon an infrastructure provided by the Nepomuk Social Semantic Desktop (SSD) [8]. This infrastructure is important for our approach since it provides the opportunity to integrate every relevant IO on the SSD into the personal semantic network of the KWer on the desktop, i.e., which represents the personal knowledge base. This integration increases the value of using the TM infrastructure and also simplifies the exploitation of KWer activities.

2.1. Architecture – Nepomuk semantic infrastructure for KASIMIR

The KASIMIR architecture is depicted in Figure 2. In the following we describe the role of the individual components compiled there.

The *Nepomuk Semantic Desktop infrastructure* [9] enables personal information management on the desktop. In this scenario the Aperture Data Wrapper [10] and the

Metadata Repository [9] embody the applied service subset of the infrastructure. These services communicate using the *Desktop Service Infrastructure*.

The *Aperture Data Wrapper* crawls the desktop for document-like IOs, such as files, emails, contacts and further objects and puts metadata references of them into the central Metadata Repository.

The *Metadata Repository* is a RDF repository. It stores the results of the Aperture Data Wrapper. Furthermore, it contains the ontologies of the personal knowledge base. This includes the Task Model Ontology (TMO) [11], i.e. describing how to store tasks and relate them to structuring and document-like desktop IOs. The Personal Information Model Ontology (PIMO) [12] formally represents the structures of a KWer's personal knowledge base on the desktop. This includes representations of document-like IOs as well as structuring IOs and their relations among each other as explained in detail in the next section. It represents the subjective view on the world from the KWer's personal perspective. The Metadata Repository stores as well the corresponding instances of the personal knowledge base.

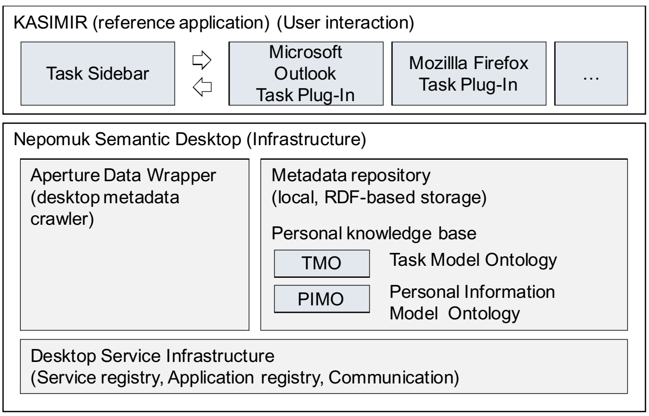

Figure 2. KASIMIR architecture.

The KASIMIR prototype bases on the above mentioned components. The task sidebar presents the tasks to the KWer that it retrieves from and stores in the RDF Repository. The task plug-ins also communicate with the RDF Repository to obtain and store task information. As well, they communicate with the task sidebar which exposes a REST-based Web Service enabling a remote control of the task sidebar from the task plug-ins.

The KASIMIR prototype has been designed for extensibility. On the one hand, further plug-ins can be developed to cover additional TM scenarios where tasks can be integrated into further desktop applications. For example this enables creating tasks from files within the file manager "Windows Explorer". On the other hand, other semantic applications can use the created semantic annotations to perform their domain-specific operations on it by accessing the RDF repository through the Desktop Service Infrastructure.

2.2. Building and using the personal semantic network on the SSD

The Nepomuk SSD establishes a semantic network throughout the personal knowledge base, i.e., the desktop, using the presented Nepomuk Semantic Desktop infrastructure. Core of this semantic network, which can be used across all desktop applications, is a personal information model. It represents the personal knowledge of the KWer and provides concepts of the IOs the KWer deals with in KW, such as tasks, persons and projects as well as relations among them. We refer to these concepts as structuring IOs as the KWer uses the same concepts, e.g. a particular person, across applications on the whole desktop. Moreover, the personal information model links these concepts with concepts of document-like IOs on the desktop, e.g., with concepts that represent email messages, files and calendar appointments.

Within Nepomuk, this *personal information model* is formally represented by the Personal Information Model Ontology (PIMO) [12]. It defines a personal knowledge ontology and links document-like IOs to the concepts of structuring IOs. Concepts of structuring IOs can be related to concepts of document-like IOs by ‘grounding’ them with defining a *hasOccurrence* relation to the document-like IO [12]. For example, a concept person ‘Claudia Stern’ can be grounded by a Microsoft Outlook contact or a concept topic ‘knowledge management’ by a website URL. As well, the *concepts* of the personal information model can be related formally within the semantic network to represent their relationship. For example, a task can be related to a person because the person works as a collaborator on the task.

The task concept and its relations to IOs on the desktop are especially relevant for the support of TM. The Nepomuk task model [11] defines the particular relations of a task to related concepts of structuring and document-like IOs as well as respective properties in a TM model. The task model ontology (TMO) [11] is the formalized representation thereof. It serves as ontology for tasks in the presented TM prototype and uses the PIMO for relations with structuring and document-like IOs involved.

3. Implicit creation of metadata in the knowledge work process

Semantic Web applications in real-life enterprise environments suffer from the problem that KWs don’t invest effort into annotation due to the missing direct benefit for them, even if from an organizational perspective contributing makes sense and helps to build up and maintain the organizational knowledge base. According to the theory of the social dilemma [13], the rational behavior of an individual KWer in a situation of uncertain benefit is to withhold contributions.

In this section we explain, how the presented TM prototype addresses such settings. By exploiting the observed actions of a KWer, which emanate from the knowledge work process, the system creates metadata according the meaning of the KWer’s action.

Using the presented mechanism to immediately exploit a KWer’s actions for metadata generation, high quality metadata is added to the knowledge base. This helps to increase the personal and the organizational knowledge base without forcing the KWer to weigh up between costs and benefits of managing annotations explicitly.

In the following, we first give an example, how the presented KASIMIR application realizes this approach to create task metadata. Then we present the underlying mechanism and give further examples.

3.1. Example - Creating tasks from within the knowledge work process

Using the example of creating a task from an email, we demonstrate how semantic annotations are generated from the corresponding KW process using an email client. Let us consider the following example. Claudia is a manager and needs to coordinate the writing of a white paper. To this end she has to find colleagues who help her in this task. To plan and execute this activity she creates a task. The trigger for this is an email from her director. For this, a right click on this email opens a menu from which she creates the task using the KASIMIR Outlook task plug-in.

This plug-in extracts email information such as subject, body, sender, and receiver, and invokes a "add task" dialogue presenting this information as pre-filled task information to Claudia. In this pre-populated "add task" dialogue she can decide about the actual email properties that are added to the task. The "add task" dialogue is optional as the task only requires a name as mandatory task property, but helps the KWer to add additional information when needed.

Presenting the pre-filled task information requires the task plug-in to process the email information and resolve it to corresponding structuring and document-like IOs on the Nepomuk platform. For example, the email itself represents a document-like IO Email and is recommended as task attachment. The task plug-in resolves the URI of the email IO using a query to the RDF repository. If the email hasn't yet been crawled by the Aperture Data Wrapper, the task plug-in triggers the Aperture data wrapper to load the particular email into the RDF repository. Persons involved in the email, i.e., obtained from the To, CC, BCC properties, represent a structuring IO Person and are recommended to the KWer as potential task collaborators or interested person. A Person is resolved using a query to the RDF repository using the email address as main key.

The task plug-in invokes a dialogue presenting this information to Claudia. She can add additional task information, such as a due date or assign a tag to the task using e.g. the IO Topic "white paper". Furthermore, she can confirm or delete the pre-populated task information. The TMO describes which role these IOs play for the task. The semantic relations are created accordingly, see Figure 3. The email IO is added as task attachment using the hasAttachment relation. For example, when approved, the recommended Persons are related to the task using a hasCollaborator relation. For Persons, that do not already exist, the corresponding IO concept is created in the PIMO.

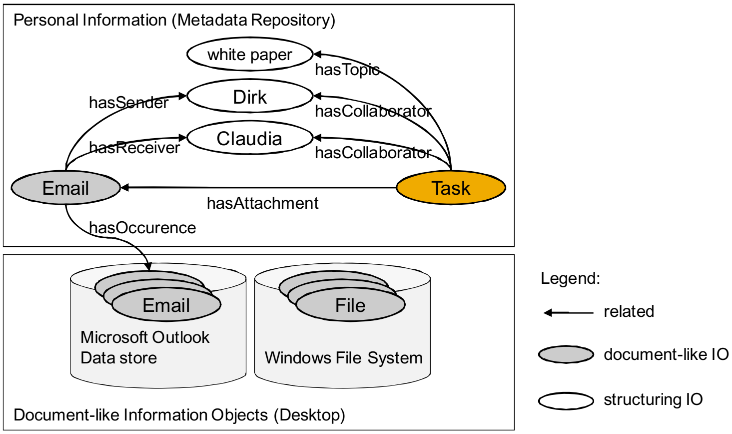

Figure 3. Example for semantic relations between tasks and further IOs.

The information integration between different semantically-enabled applications allows for a *cross-application usage of the semantic information* of the Nepomuk Semantic Infrastructure. For example, the IOs of persons, emails, and topic are used not only by the task management application. By using the Nepomuk Semantic Infrastructure, other Nepomuk services can access the Metadata Repository and use these IOs. Moreover, they can add their own information, e.g., using annotations. These annotations can then in return be used by the task sidebar. They consists either of a generic annotation, like e.g. as defined with the Nepomuk nao:hasAnnotation relation [14] or by another special purpose relation like e.g. for a person a community recommender ranking of the person.

3.2. Mechanism principle and further examples

The previous section explained in detail, how the KASIMIR task prototype creates task metadata from the KW's action of pressing a button in the email application. In this section, we present an abstracted principle of the underlying mechanism. The principle of the mechanism is depicted in Figure 4 and explained in detail below.

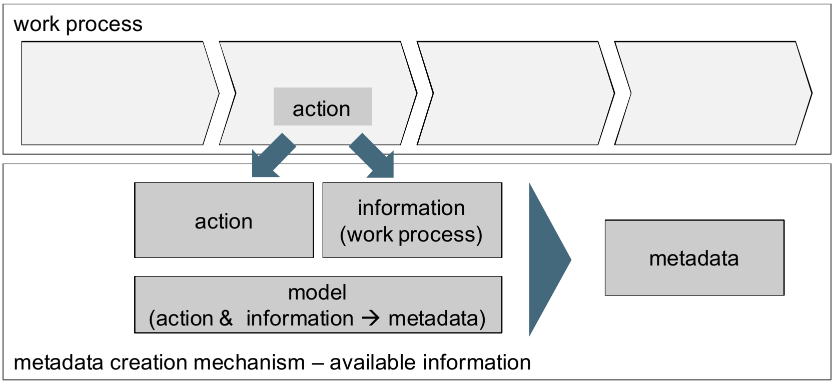

Figure 4. Principle of seamless metadata creation mechanism.

To implicitly create metadata out of a KWer's work process, a particular *action* performed by the KWer within this process serves as the trigger for the mechanism. Usually, KWers perform such actions as part of their work, i.e., there is no extra action needed for the described mechanism. In the presented example, we integrated a plug-in into an existing application to capture the KW's action of turning an email into a task.

When the KWer performs this action, there is related *information* available that can be used for the metadata creation. This information comes on the one hand from the performed action and the context of the work process as well as on the other hand from IOs involved in this action. In the given example, such information was contained in the email header of the involved email.

A *model* defines how the triggering action and the available information are transformed into metadata. In particular, it is possible that the model describes how to obtain additional information that enriches the existing information. This metadata creation mechanism explicitly goes beyond simply transforming attributes of the respective IO into metadata since the model can include advanced reasoning methods that, for example, can connect these attributes with the respective action to derive additional metadata. The next example illustrates this point.

Numerous KWers regularly perform an "ordering action" on their incoming emails, i.e., they file their emails into specific folders to keep a good overview. The model uses the meaning assigned to the email folder by the KWer, e.g., a folder called "follow up" would have the meaning "follow up within two weeks", and combines it with the available email information. Then, for example, sorting an email into this folder could lead to the creation of a task with the respective due date. The creation of further metadata, e.g., the task is marked with the category "important tasks", could be additionally defined by the model. This metadata creation happens implicitly when the KWer performs her regular email ordering actions.

Another example of creating extended metadata would be in a case of an email folder called "important". Here, e.g., the specific email would be categorized with the label "important" as well as the email text is used to rank contained words of the email with a higher importance in the desktop search engine to optimize email retrieval and further applications that leverage email information on the SSD.

The created metadata is available for use by other applications on the desktop as it is located in the common Metadata Repository. As well, this metadata can be shared among several KWers, with the prerequisite of taking into account possible security and privacy concerns.

4. Evaluation – KASIMIR task plug-ins well received

We evaluated the KASIMIR prototype to find out about whether the KASIMIR prototype implementation meets the above presented requirements, i.e., the KWers' support needs. We focus on the evaluation of those aspects of task management as implemented with the desktop application task plug-ins for Mozilla Firefox and Microsoft Outlook.

The methodology chosen for the evaluation of the KASIMIR prototype was a formative usability evaluation using the think-aloud protocol [14]. We evaluated the

Kasimir prototype with eight participants, seven men and one woman. Each evaluation session took approximately one hour. The medium age of the participants was 33 years and it varied from 26 years to 52. All informants were employees of a large business software producer working with, e.g., project management, ontologies, semantics, security, access control, system architecture, business grids, and software development. The evaluation sessions consisted of a pre-interview, a task phase and a post-interview. The sessions had one moderator and one or more persons observing. The evaluations were videotaped for further analysis. All evaluation sessions were individual.

Both *KASIMIR task plug-ins were well received* despite some user interaction problems. The evaluation showed that the informants liked the idea of task plug-ins in applications like email clients and web browsers, especially the Outlook extension was received positively. They wanted to add resources to a task. As well, they liked the idea to interact with the tasks from other applications. They could see that task plug-ins could be useful in other applications, e.g., PowerPoint and Word.

However, there were problems with how the tasks were managed in the task plug-ins. In particular, the informants were confused about the relation between the tasks shown on the one hand within the task plug-ins and on the other hand in the task sidebar. One informant pointed out that there were two “work-centers”, one in the extension and one in the Kasimir application. He wanted one “work-center”. One informant, that was more negative to the extensions, said that he would rather create a bookmark folder than tasks in the web browser.

Summing up we can say that although the functionality in the task plug-ins was yet unfamiliar to the informants and led to some confusion, many informants liked the idea to connect resources to a task and that they liked the idea to interact with the tasks from other applications.

5. Discussion and further work

We have presented the KASIMIR prototype that integrates TM activities into the desktop applications of knowledge workers. Thus it allows for the implicit creation of semantic annotations that help building a personal semantic network. At the same time it provides a consolidated overview of tasks across application boundaries. Due to its application integration KASIMIR demands only minimal effort from KWers to maintain tasks. Due to its rooting in semantic web technologies, it considerably reduces the major bottleneck of existing semantic applications in an enterprise environment, namely the costs of metadata creation.

If we look for similar user-centric approaches, we find a parallel to the approach of Adali et al. [2], who also extract semantic annotations from task data. However, they do not extract them directly from the operating TM system but by mining. This introduces an additional mediation process that reduces the metadata quality.

The design goal for the presented prototype was to provide a work-embedded application that supports the KWer in performing task management. At the same time, the approach leverages the KW’s actions to create task metadata. The prototype intentionally doesn’t look like a semantic application, although it contributes to the

semantic desktop in the described way. In the next versions it will even enable the browsing of the semantic information.

Further work consists of implementing the presented examples of the mechanism as well as evaluating the mechanisms in more detail. The performed evaluations revealed numerous ideas for further directions on the development of the task plug-ins and the task sidebar. Being based on the Nepomuk Semantic Middleware, the KASIMIR prototype stands to further benefit from enhancements of this middleware platform. Introducing further domain ontologies, e.g. about employees and organizational structure, to the existing Nepomuk ontologies will leverage existing enterprise knowledge.

Finally, if we look at the opportunities for the use of ontologies that are opened by the presented approach, we see a promising way to generate the required metadata in a cost-efficient way. This is an essential precondition for the Social Semantic Desktop to be filled with semantic contents necessary to exploit the opportunities provided by this new technology. The Task Management can particularly benefit from it. On the one hand, it is the central medium to gain this metadata and, on the other hand, is the application that mostly benefits from it. To understand the latter point we simply have to recall why task management applications are mainly ignored by knowledge workers today [11]. The reason is that these applications do not enrich the basic task information and thus do not provide much benefit. Semantic technologies based on ontologies as they have been developed in Nepomuk can provide such enrichment and increase the value of TM for knowledge workers in this way.

References

[1] U. Cress, B. Barquero, B. Buder, FW Hesse: Social dilemmas in knowledge communication via shared databases, In Barriers and Biases in Computer-Mediated Knowledge Communication – and How they may be Overcome (R. Bromme, FW. Hesse, H. Spada, Eds), Springer, New York, (2005) 143–167.

[2] S. Adali, S. Pearce, M-L. Sapino: On Querying and Exploring Activities on a User's Desktop, 22nd International Conference on Data Engineering Workshops (ICDEW'06) (2006) x103.

[3] V. Bellotti, N. Ducheneaut, M. Howard, I. Smith: Taskmaster: recasting email as task management, Workshop: "Redesigning Email for the 21st Century", CSCW, 2002.

[4] M. Czerwinski, E. Horvitz, S. Wilhite: A diary study of task switching and interruptions, Proceedings of the SIGCHI conference on Human factors in computing systems, Vienna, Austria, ACM Press, (2004) 175–182.

[5] U. V. Riss: Knowledge, Action, and Context: Impact on Knowledge Management. In: K.-D. Althoff, A. Dengel, R. Bergmann, M. Nick, T. Roth-Berghofer (Eds.): Professional Knowledge Management, Lecture Notes in Artificial Intelligence, (2006), vol. 3782, 598-608.

[6] V. Bellotti, B. Dalal, N. Good, P. Flynn, D.G. Bobrow, N. Ducheneut: What a to-do: Studies of task management towards the design of a personal task list manager, In Proceedings of CHI 2004, ACM Press, 2004.

[7] S. Zhang, C. Zhao, P. Moody, Q. Liao, Q. Zhang: Lightweight Collaborative Activity Patterns in Project Management, Engineering Psychology and Cognitive Ergonomics, (2007) 465-473.

[8] T. Groza, S. Handschuh, K. Moeller, G. Grimnes, L. Sauermann, E. Minack, C. Mesnage, M. Jazayeri, G. Reif, R. Gudjonsdottir: The nepomuk project - on the way to the social semantic desktop. In T. Pellegrini and S. Schaffert, editors, Proceedings of I-Semantics' 07, JUCS, (2007) 201–211.

[9] Nepomuk Project: Deliverable D6.1, First version of the Backbone and Connector Infrastructure, 2006, http://nepomuk.semanticdesktop.org/xwiki/bin/view/Main1/D6-1.

[10] Aperture. http://aperture.sourceforge.net.

[11] Nepomuk Project: Deliverable D3.1, Task Management model, 2006, http://nepomuk.semanticdesktop.org/xwiki/bin/view/Main1/D3-1.
[12] L. Sauermann, L. van Elst, A. Dengel: Pimo - a framework for representing personal information models, In T. Pellegrini and S. Schaffert, editors, Proceedings of I-Semantics' 07, JUCS (2007), 270–277.
[13] R.M. Dawes: Social dilemmas, Annual review of Psychology (1980), Vol. 31, 169-193.
[14] Scerri, S., Sintek, M., van Elst, L., Siegfried Handschuh: NEPOMUK Annotation Ontology Specification, 2007, http://www.semanticdesktop.org/ontologies/nao/.
[15] J.S. Dumas, J. Redish: A Practical Guide to Usability Testing, Westport, CT: Greenwood Publishing Group Inc., 1993.

Formal Ontologies Meet Industry
S. Borgo and L. Lesmo (Eds.)
IOS Press, 2008

Roles: A Four-Dimensional Analysis

Matthew West[1]
Reference Data Architecture and Standards, Shell International Petroleum Co. Ltd, UK

Abstract. In ontology the underlying nature of things is sought. Thus it is that the roles things play can sometimes be neglected. Some different sorts of things that get called roles: places in relations, ways of participating in activities, and ways of being a part of a system, are considered. Roles are then analysed from a 4-dimensional and extensional perspective. We find that in individuals a role is a state of some individual that participates in a relationship or an activity, whereas with classes it is the class itself that participate in a place in a relation. In systems (both social and functional roles) we find that roles are replaceable parts of the system that are the purpose the part plays in the whole. A 4-dimensionalist ontology of roles, linked to an upper ontology is presented, and this is used to set out the key properties of roles as found here and by others.

Keywords. ontology, 4-dimensionalism, role.

Introduction

In ontology the underlying nature of things is sought. Thus it is perhaps not surprising that the roles things play has been, or at least seemed to the author, to have been relatively neglected or thought to be insubstantial. Further, the work that has been done presents divergent views on the nature of roles.

In this paper I examine some different sorts of role, and develop a 4-dimensional[2] analysis of them including a 4D ontology linked to an upper ontology. This is then used to set out the key properties of roles, and is compared to the properties and questions set out by others [11],[12].

1. Roles Found in Literature and Industry

A particularly good survey of the literature on roles can be found in Steimann [11], and a significant study has been done by Loebe [3].

In the literature, a number of different sorts of role are recognized:

1. Relational Roles (e.g. Loebe [3], Masolo et al [3], Sowa [5])
2. Participating or Processual Roles (e.g. Loebe [3], Sowa [1])
3. Social Roles (e.g. Loebe [3], [12] Masolo et al [6], Smith [7], Mizigouchi [13])
4. Functional Roles (e.g. ISO 15926 [8])

Steimann [11] has identified a number of properties of roles, which Loebe [12] has developed further, but rather posing questions. These are presented in Table 1.

---

[1] Corresponding Author: Matthew West, DFD/D21, Shell International Petroleum Co. Ltd., Shell Centre, London, SE1 7NA, United Kingdom; Email: mail@matthew-west.org.uk.

[2] A good description of 4-dimensionalism can be found in [1] and [2].

Table 1: Properties of roles as identified by Steimann and Loebe

Steimann	Loebe
1. A role comes with its own properties and behaviour. 2. Roles depend on relationships. 3. An object may play different roles simultaneously. 4. An object may play the same role several times, simultaneously. 5. An object may acquire and abandon roles dynamically. 6. The sequence in which roles may be acquired and relinquished can be subject to restrictions. 7. Objects of unrelated types can play the same role. 8. Roles can play roles. 9. A role can be transferred from one object to another. 10. The state of an object can be role-specific. 11. Features of an object can be role-specific. 12. Roles restrict access. 13. Different roles may share structure and behaviour. 14. An object and its roles share identity. 15. An object and its roles have different identities.	1. Roles as individuals vs. roles as universals [1,14,15][3]. Do role individuals exist or are roles a specific kind of universals? 2. Role identity [14,15]. Do roles have an identity different from their players? 3. Dependence, relational nature of roles, and contexts [2]. In which ways do roles depend on other entities? 4. Roles with own properties and behavior [1,11]. Do all roles come with their own properties and behavior? 5. Dynamicity and anti-rigidity [4,5,6,9]. In which way are roles considered "dynamic"? Does anti-rigidity apply to all roles? 6. Role-playing roles [8,9]. Can roles play roles? What relations among roles exist in general? 7. Multiplicity of roles [3,4,7,9]. To how many roles of how many types can natural universals be linked? How many roles can be played by instances of natural universals? 8. Generalization hierarchies with roles [13]. How can role and non-role terms be arranged in a single generalization hierarchy? 9. Role abstraction and complementary roles. Why is abstraction among relational and processual roles reasonable? What should complementation mean on the universal level? 10. Pure roles. What is the difference between roles like child and son? 11. Integrating roles with qua-individuals. Should role individuals be identified with qua-individuals? 12. Meta-level status of roles. From a meta-level perspective on the model presented, are role-individuals genuine entities?

Some questions about which there does not seem to be consensus are:

1. Are these different sorts of role (a particular role playing) individuals or classes, and if individuals, abstract or concrete?
2. Are role types, specializations of base types or not (e.g. is employee a specialization of person?)
3. Is a role identical to the object playing the role (see Steimann {14,15} in Table 1.

2. Roles in Relationships

I want to start by distinguishing between how something is represented, and what something is. For example, it is quite possible for an activity to be represented by a

[3] The numbers refer to the numbering of the Steimann properties.

relation, with each of the roles played in the activity represented by a place in the relation. This, however, does not make an activity a relation. An activity is an individual that causes change. On the other hand, a relationship is something static, that holds for the period of its validity. Again, a relationship may, or may not, be represented by a relation (an alternative for example would be to represent the relationship by a class, and the roles played in the relationship by relations). Here I am talking about relationships and activities, rather than relations.

In the 4-dimensional paradigm with extensionalism, the primary objects are either spatio-temporal extents, or classes, where each has its extension as its basis for identity. Thus I will consider three types of relationship:

1. Relationships between individuals.
2. Relationships between an individual and a class.
3. Relationships between classes.

2.1. Roles of Individuals in relationships

I shall start with a space-time diagram of ownership as an example relationship between individuals, **Figure 1**.

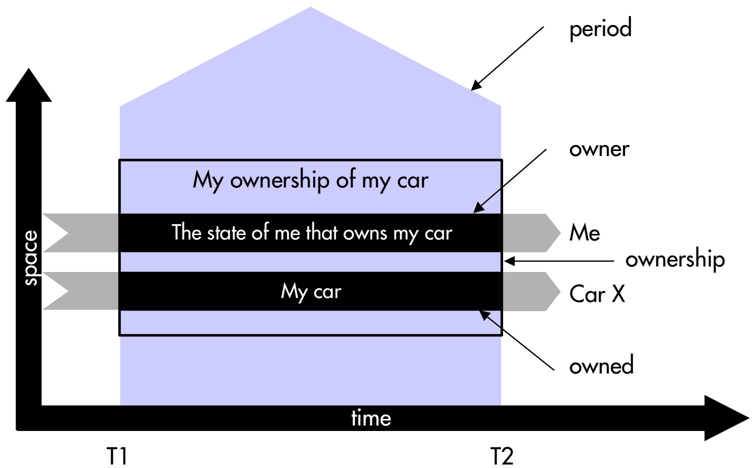

Figure 1. A space-time map for a relationship between individuals

The figure illustrates the ownership of my car by me. The grey chevrons represent me for the whole of my life and car X for the whole of its life, the chevrons indicating that the start and end are beyond the limits of this diagram. The black rectangles represent the state of me that owns my car and the state of car X owned by me (my car) respectively, and the empty black box represents the aggregation of those states that together constitute the ownership relationship. Notice that the relationship is itself a spatio-temporal extent. The arrows show a classification of each state by the appropriate class, in the case of the black rectangles these are classes of role. The light grey vertical chevron represents the period that the relationship is a spatial part of. Notice how much of what one might ordinarily think of as a simple relationship is in fact made up of whole-part relationships.

A special case of this pattern is whole-part. This is illustrated twice over in **Figure 1**, as the owned and owner roles are parts of the ownership relationship, and in turn the ownership relationship is part of the period that the ownership occurs in.

What you will notice here is that a complex relationship between individuals has been decomposed into a number of either classification or composition relationships. I conjecture that this is the case for all non-primitive contemporaneous relationships between individuals (an example of a non-contemporaneous relationship would be one between a history teacher and Napoleon when the teacher was talking about Napoleon). Masolo [4] and Kozaki [10] find similar results.

2.2. Roles in Relationships Between Individuals and Classes

Figure 1 also illustrates an example of a relationship between a class and an individual, that of classification, represented by the arrows in the figure. Classification is a special case, since it is a primitive relationship that is required to be able to say anything. However, it illustrates the nature of relationships between individuals and classes. The key element is that of temporality. The relationship is between the state of the individual (role) for which the relationship is true and the class. In the case of classification, this supports the extensionalism of classes, since rather than the membership of a class changing over time, the states, which incorporate the period of validity, are simply members of the class, which is then an eternal statement. Although classification is a special case, this pattern holds for other relationships between individuals and classes.

2.3. Roles in Relationships Between Classes

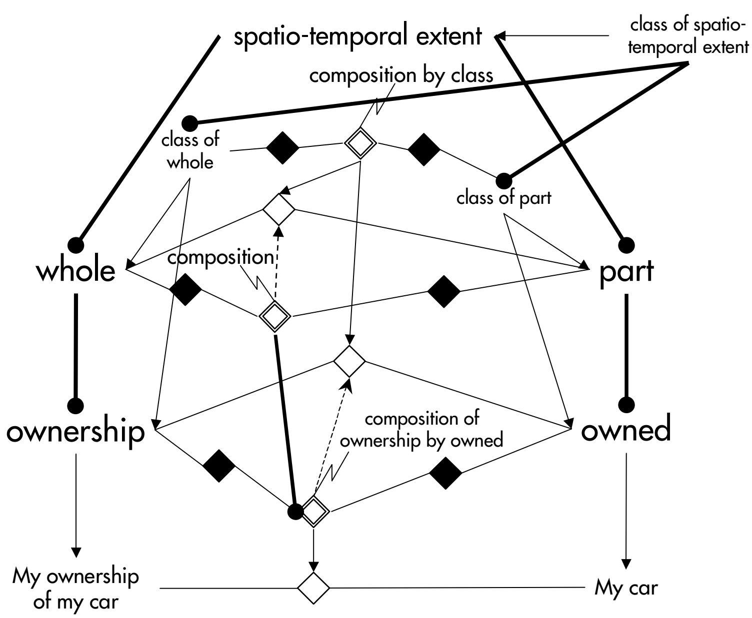

Figure 2. An illustration of relationships between classes

Figure 2 gives examples of relationships between classes. Building on the example above. The notation is as follows[4]:

- A single line diamond represents a simple relationship, with lines linking to the related objects,
- A double diamond represents a class of relationship, i.e. a set of relationships,
- A black diamond represents a relationship between a class of relationship and a class of role that participates in member relationships,
- The dashed arrow represents the implication that a relationship exists between the roles related to the class of relationship,
- The thick lines represent a specialization relationship, with the subclass at the lollipop end,
- The lightning strike is a call out to the name of the object.
- As in **Figure 1** an arrow represents a classification relationship, with the member at the arrowhead end.

Starting at the bottom of the diagram we find the relationship between my ownership of my car, and my car. This relationship is classified as a composition of ownership by owned relationship. This class in turn is related to the classes ownership and owned, indicating they are the classes of role played in the composition of ownership by owned relationships. Moving up the diagram again, we see that composition of ownership by owned is a specialization of composition, and that ownership is a specialization of the class of role, whole, and owned is a specialization of the class of role, part. Finally, we can see that both whole and part are specializations of spatio-temporal extent.

Now in addition, there is a relationship between the classes ownership and owner, and another one between whole and part. These are both implied by the classes of relationship at the other end of the dotted arrow. Now with a relationship between classes, it is the classes themselves that play the role. This is because classes are timeless, so if the relationship is true it is always true. Just as with the relationship between individuals, there is also a class of relationship for the relationship between classes. This is composition by class, and is shown near the top of the diagram. The relationships above are shown as members.

3. Roles of Individuals in Activities

Figure 3 illustrates the way that roles occur in an activity. This example is for the replacement of a pump impeller. The notation is the same as in **Figure 1**. First someone makes a request in the engineering department to someone in the purchasing department of the customer organization for a new impeller. Then the purchasing department of the customer organization places an order with someone in the sales department of the supplier organization. The sales organization then requests delivery of the impeller to the customer organization by delivery, who executes the delivery which is received by the customer organization. The impeller is then installed (details not shown).

The roles in the activity are the states of the objects whilst they are participating. The activity consists of the roles participating in the activity. It is noticeable that the

[4] This notation is based on that used in [8] to illustrate instances of the data model.

pattern of roles in activities is the same as the pattern for relationships between individuals. The difference between them is that an activity causes change, and the participations in the activity are themselves activities, whilst a relationship is about something constant.

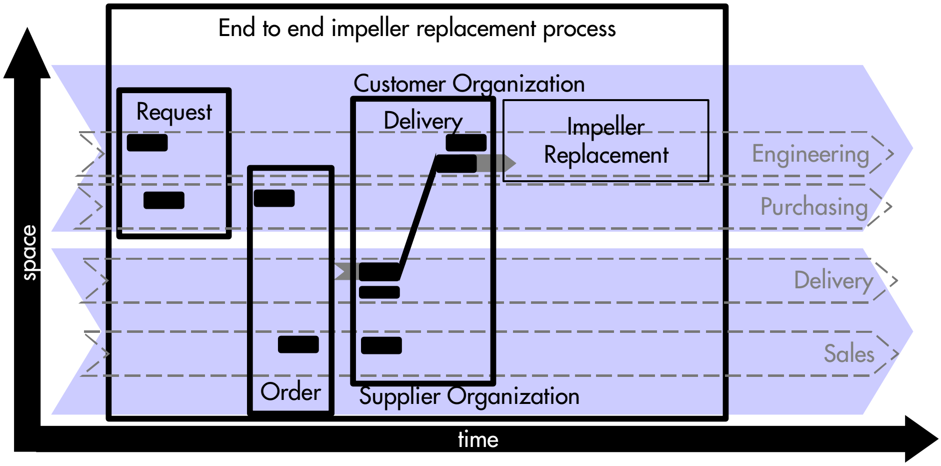

Figure 3. An example of roles in an activity

4. Roles as Replaceable Parts in Systems

As we shall see, in a 4-dimensional analysis, social roles and functional roles share the same pattern, so, whilst considered separately, they are brought together under a more general heading.

4.1. Social Roles

A social role is a role in human affairs that is defined independently of the person who fulfills the role, and, from time to time, the person who fills the role may change. Examples include the Lord Mayor of London, the President of the United States, and the Secretary to the Head of Department in the School of Computing at the University of Leeds.

A 4-dimensional analysis of social roles was presented in [11]. The example used there is illustrated in **Figure 4** below.

The figure shows that Bill becomes president for a period of time, i.e. a state of Bill is also a state of the President of the United States. When Bill leaves office George replaces him, and he is then president for a period of time, i.e. a state of George is also a state of the President of the United States. There are four things to note here:

1. That a social role consists of the temporal parts of those who play the role whilst they are playing it,
2. That a social role can change all its parts at once, and survive that change, unlike ordinary physical objects,

3. That a social role can go through a period of non-existence, if for example there is a gap between one person leaving office, and another taking the office up,
4. If there is no United States, there is no President of the United States, i.e. the social role is dependent on the social system that it is a part of. (Mizigouchi [13] also recognizes the dependence of social roles, but only puts that dependence on a context, rather than explicitly recognizing it is on the social system it is part of.)

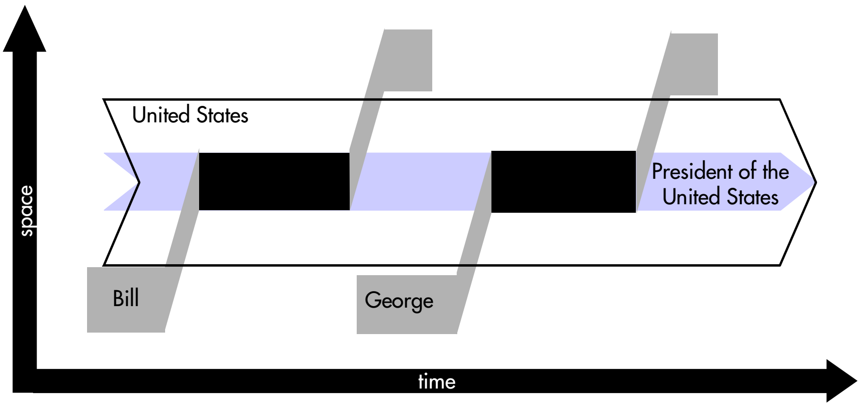

Figure 4. Space-time map for President of the United States[5]

A consequence of this is that it makes sense to talk about shaking the hand of the President, because the President is a real physical object, though not an ordinary physical object.

4.2. Functional Roles

A functional role is a part of a functional object, that can be wholly replaced, and yet retain identity. Examples are: the offside front wheel of a car, the port engine of a plane, or the pump at the bottom of a distillation column on a process plant.

This last example, taken from [8] and [9] is illustrated in **Figure 5** below.

The Crude Distillation Unit has a pump at the bottom of a distillation column, labeled P101. A pump, with manufacturers serial number, Pump 1 is installed as P101, and operated as such by the plant operators. They are concerned with the availability of P101, not which pump is installed there. However, the maintenance engineers record maintenance against Pump 1. Now at some point in time, Pump 1 breaks down and is replaced by Pump 2. The operators continue to operate the replacement pump as P101. Maintenance repair Pump 1 and perhaps install it in some other process plant performing another duty.

As with the social role, things to note are:

1. That a functional role consists of the temporal parts of those physical objects that play the role whilst they are playing it,

---

[5] Whilst this figure shows the general pattern, where a social role my go through a period of non-existence between one incumbent leaving office, and another taking office, in the case of the President of the United States there are rules such that whatever should happen to the present incumbent, there is always someone who is the President.

2. That a functional role can change all its parts at once, and survive that change, unlike ordinary physical objects.
3. That a functional role can go through a period of non-existence, if for example there is a gap between one physical object being removed, and another being installed.
4. The existence of the functional role (P101) is dependent on the existence of the Crude Distillation Unit.

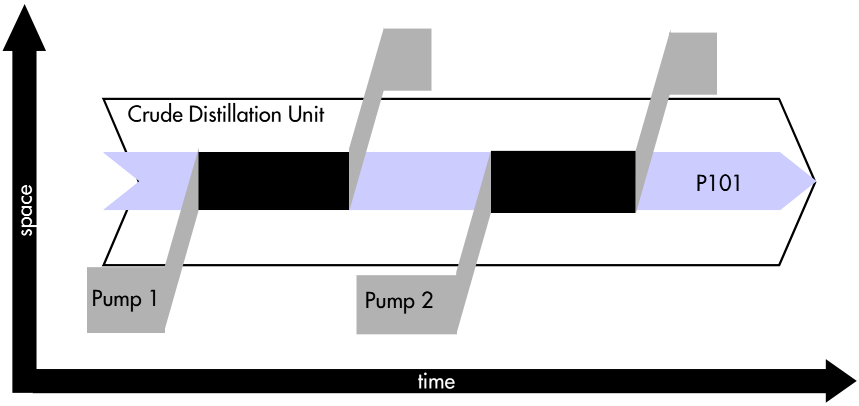

Figure 5. The replacement of a pump in a process plant

It seems to be particularly difficult to come to terms with periods of non-existence. How can something not exist when I can talk about it? This is understandable, however, we do not find it difficult to talk about historical things that do not exist now. Perhaps a better way to understand temporary non-existence is a practical one. If you are a plant operator, and you are asked to start up P101 when there is nothing installed there, you will not be able to do it. The reason you will not be able to do it is because at that time it does not exist.

4.3. Similarities and Differences Between Functional and Social Roles

The general pattern of social roles and functional roles is clearly the same. The difference is just the level of reality at which the system operates. Indeed, one can look at other levels of reality and see that systems with replaceable parts can exist at those levels too. A particular case in point is biological systems, where an organ is a replaceable part of a living creature.

5. A 4-Dimensionalist Upper Ontology of Roles

Having analyzed with examples the spatio-temporal patterns of different sorts of role, I now present an ontology based on that analysis as an extension to a 4 dimensionalist upper ontology. This is shown in **Figure 6** below. The thick lines show subtype/supertype relationships with the subtype at the lollipop end. Thin lines show relations, which, when the line is solid are mandatory at the sharp end. Relation names are read from the sharp end to the lollipop end.

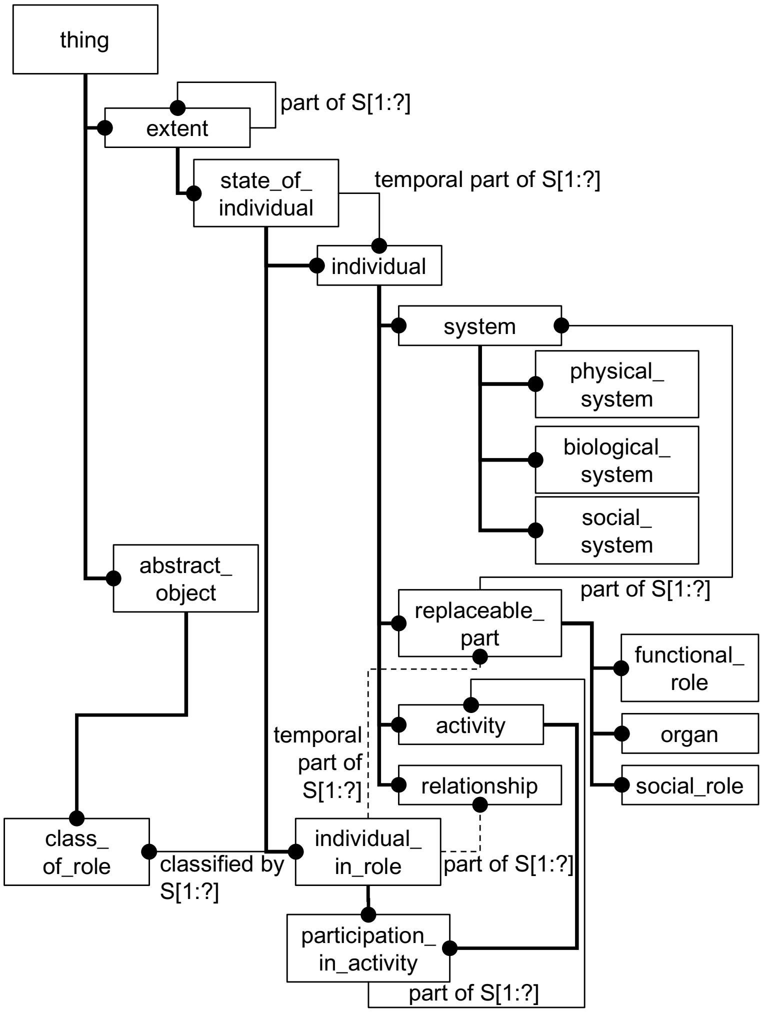

Figure 6. The 4-dimensionalist basic upper ontology for roles

A **thing** is anything that exists, real or imaginary. A thing may be either an **abstract object** or an **extent** (but not both). **Abstract object**s are either relations or classes (only a subtype of class is shown here). An **extent** is any spatio-temporal extent, i.e. any arbitrary piece of space-time, which does not even have to be contiguous.

An **individual** is an **extent** that is something for the whole of its life, the ordinary objects and activities that we recognize. The subtypes of **individual** considered here are:

- **system**, which is an organized or connected group of objects, and may be a **physical system**, a **biological system**, or a **social system**;
- **replaceable part**, which is a *part of* at least one system, and may be a **functional role**, an **organ**[6], or a **social role**;

[6] **Organ** is only an example of a biological **replaceable part**, not all biological **replaceable part**s are necessarily organs.

- **activity**, which causes change;
- **relationship**, which is an unchanging state of at least two **individual**s.

State of individual is any temporal part of an **individual**, and so **individual** is itself a subtype of **state of individual**, since it is a maximal state. An **individual in role** is a **state of individual** that is playing a role. This may be *part of* a **relationship**, or *temporal part of* a **replaceable part**. An **individual in role** is *classified by* a **class of role**, which is a subtype of **abstract object**.

A **participation in activity** is any **individual in role** that is also an **activity**, and is a *part of* an **activity**. It is the **individual** whilst they participate in the **activity**.

6. Discussion

In section 1 I noted the properties and questions that Steimann and Loebe have raised about roles. Now that I have constructed a 4-dimensional ontology of roles, I consider the consequences for these, making the following statements for the ontology I have constructed.

1. A role (**individual in role**) is a **state of individual** that is a separate object with separate identity from the **individual** playing the role. As such it has its own properties and behavior. Steimann {1,10,11,14,15}, Loebe {1,2,4,10[7],11[8],12}.
2. A role (**individual in role**) is existence dependent on both the **individual** that plays the role, and the **relationship**, **activity**, or **system** that the role is part of. Steimann {2}, Loebe {3}.
3. Since there is no restriction on one **state of individual** temporally overlapping with another, an object may play the same role many times or different roles simultaneously. Steimann {3,4}, Loebe {7}
4. A role (**individual in role**) is a *temporal part* of the **individual** playing the role. Steimann {5}, Loebe {5}[9]
5. The sequence in which roles may be acquired and relinquished can be subject to restrictions. Steimann {6}.
6. **Individual**s of unrelated type may play the same role. Steimann {7}, Loebe {7}.
7. Roles (**individual in role**) can play roles. There is nothing to prevent one **individual in role** being a *temporal part of* another, which is what it means in 4D for one role to play another. Steimann {8}, Loebe {6}.
8. For a **replaceable part**, all of its parts may be replaced, and it may go through periods of non-existance. Steimann {9}, Loebe {7}.
9. One type of role may be a subtype of another, and hence may share structure and behavior. Steimann {13}, Loebe {8,9}.
10. An **activity** or **relationship** is constructed from the **individual in role**s that participate in them. Loebe {9}.

It is particularly noticeable that very many questions raised are answered by understanding that a role is a state of the individual playing the role.

---

[7] Not all **state of individual** are **individual in role**, so child is a **state of individual** that is not a role, whilst son is an **individual in role**.

[8] Qua-individuals seem to be invoked when 3 dimensionalists really need states.

[9] Dynamicity and anti-rigidity make no sense in a 4D ontology.

7. Conclusions

Four different types of role, relational, participating, social, and functional, have been analyzed from a 4-dimensional perspective. We have seen that roles played by individuals consist of states of the individuals playing those roles. In particular, social and functional, but also participating roles can have all their parts change at once, and survive that change, unlike ordinary physical objects. We have also seen that role classes, rather than being specializations of the class of thing that plays the role, are specializations of states of those things. Finally, we have been able to show how this ontology exhibits all the properties and answers all the questions posed by Steimann and Loebe.

References

[1] Sider, Theodore Four Dimensionalism - An Ontology of Persistence and Time 2001 Oxford University Press, ISBN 0-19-926352-3
[2] Hawley, Katherine How things persist Oxford: Clarendon Press 2001
[3] Loebe, F. 2003. An Analysis of Roles. Toward Ontology-Based Modelling. Master's Thesis, University of Leipzig.
[4] Masolo, C. and Guizzardi, G. and Vieu, L. and Botazzi, E. and Ferrario, R. (2005) Relational roles and qua-individuals. In: AAAI Fall Symposium on Roles, an Interdisciplinary Perspective, 2005, Virginia, USA. pp. 103-112. American Association for Artificial Intelligence (AAAI). ISBN 978-1-57735-254-9
[5] Sowa, J.F. Knowledge Representation: logical, philosophical and computational foundations Brooks/Cole - Thomson Learning, 2000, ISBN 0-534-94965-7
[6] Masolo, C., Vieu, L., Bottazzi, E., Catenacci, C., Ferrario, R., Gangemi, A., and Guarino, N., 2004. Social roles and their descriptions. In Ninth International Conference on the Principles of Knowledge Representation and Reasoning, Whistler Canada.
[7] B. Smith, Social Objects http://ontology.buffalo.edu/socobj.htm 2002
[8] ISO 15926-2:2003, Industrial automation systems and integration - life cycle data for process plant - Part 2: Data model.
[9] West, Matthew Some Industrial Experiences in the Development and Use of Ontologies EKAW04 Workshop on Core Ontologies, 2004
[10] Kozaki, K., Sunagawa, E., Kitamura, Y., Mizoguchi, R.,Fundamental Consideration of Role Concepts for Ontology Evaluation, Proc. of Evaluation of Ontologies for the Web (EON2006) 4th EON Workshop, Edinburgh, United Kingdom, May 22, 2006
[11] Steimann, F. On the representation of roles in object-oriented and conceptual modelling. Data & Knowledge Engineering, 35(1), 83–106, 2000
[12] Loebe, F. Abstract vs. social roles – Towards a general theoretical account of roles, Applied Ontology 2 (2007) 127–158, IOS Press
[13] Mizoguchi, R. et al The model of roles within an ontology development tool: Hozo, Applied Ontology 2 (2007) 159–179, IOS Press

Formal Ontologies Meet Industry
S. Borgo and L. Lesmo (Eds.)
IOS Press, 2008

Using Background Knowledge and Context Knowledge in Ontology Mapping

Hassan Fatemi, Mohsen Sayyadi and Hassan Abolhassani
Semantic Web Research Laboratory,
Computer Engineering Department,
Sharif University Of Technology, Tehran, Iran
{fatemi,sayyadi}@ce.sharif.edu;abolhassani@sharif.edu

Abstract. Recent evaluations of mapping systems show that lack of background knowledge, most often domain specific knowledge, is one of the key problems of mapping systems these days. In fact, at present, most state of the art systems, for the tasks of mapping large ontologies, perform not with such high values of recall ($\sim$ 30%), because they mainly rely on label and structure based similarity measures. Disregarding context knowledge in ontology mapping is another drawback that almost all current approaches suffer from. In this paper we use the semantic web as background knowledge and introduce a novel approach for capturing context knowledge from the ontology for improving mapping results. We have successfully tested our approach with the public test cases of the Ontology Alignment Evaluation Initiative 2005 and achieved promising results.

Introduction

It is hard to come up with one single universal shared ontology which is applauded by all players in the semantic web realm. In other words, distributed and heterogeneous nature of the semantic web needs different ontologies. It seems clear that ontologies face the same or even harder problems with respect to heterogeneity as any other piece of information. The heterogeneity problem undermines the main goal of ontologies and even semantic web which is establishing interoperability between different agents and services. The attempts to improve system interoperability will therefore rely on the reconciliation of different ontologies used in different systems, so interoperability among applications in heterogeneous systems depends critically on the ability to map between their corresponding ontologies.

The problem of ontology mapping (also known as ontology integration, semantic integration, ontology matching, etc.) has a central role in the development of knowledge based systems. Therefore, we need ontology mapping techniques for creating a set of axiomatic bridges between the ontologies. Various solutions for ontology mapping have been proposed so far, see for recent surveys[1, 2]. Today, mapping between ontologies is still largely done by hand, in a labor-intensive and error-prone process. As a consequence, semantic integration issues have now become a key bottleneck in the deployment of a wide variety of information management applications. Hence, improving the results of existing mapping approaches would be of high value.

Another drawback of existing mapping approaches is that most of them generally derive correspondences between terms, eventually associated with a confidence that aggregates different similarity values. For example, during the 2005 Ontology Alignment Contest (OAC)

[Euzenat et al., 2005a] only one algorithm (CtxMatch [Bouquet et al., 2005]) was able to go beyond equivalences and also produce partial relations in the form of subconcept relations. In this year, all the tools produced only weighted correspondences [Euzenat et al., 2006], thus failing to provide semantic mappings that would be directly exploitable in dynamic scenarios as machine interpretable relations. We will discuss semantic mappings later in more details.

The rest of the paper is organized as follows. In section 2 we discuss about main ideas of some related works and enumerate their imperfections and shortcomings. In section 3 we introduce the role which context knowledge plays in ontology mapping and also introduce a novel approach to capture it. Section 4 briefly comments on background knowledge, most specific the Semantic Web, and discusses its features. Section 5 is devoted to the implemented application to test the applicability of our idea and section 6 evaluates the implemented application. The final section presents conclusions and recommendations for future work in this area.

1 Related Works

[3, 4] use the background knowledge concept in an interesting way. They proposed a method for aligning two lists of terms using structure-rich ontologies as background knowledge. There already exist several techniques for ontology alignment. However, the existing techniques are most effective when the ontology contains structure. If the vocabularies have no structure, and only consist of lists, these approaches fall back on lexical techniques only. These two approaches, in the first step, establish relationships between the concepts in the two lists to be mapped and a structured source of background knowledge. Through this relationship, the concepts in the lists acquire properties with values taken from the background knowledge. From this background knowledge a mapping can be induced between the concepts in the list by the heuristic that if two concepts in the lists have common properties and if they have related values for these properties, then these two concepts are related.

Analogous to the above [5]uses a comprehensive ontology as background knowledge for mapping concepts of two ontologies.

The major imperfection of the two above mentioned approaches that rely on a comprehensive ontology for mapping is that the appropriate comprehensive ontology needs to be manually selected prior to matching. In many scenarios this approach is unfeasible as we might not know in advance which terms from which ontologies need to be matched. Even in the cases where a reference ontology can be manually selected prior to matching, there is no guarantee that such an ontology actually exists. Indeed, many domains lack large, richly formalized domain ontologies that would cover the ontologies to be matched.

[6], proposed a method for taking account of background knowledge in order to address the issue. Focusing on tree-like structure of ontologies it thinks of a concept of a label as the propositional formula which stands for the set of documents that one would classify under the label it encodes and concept at a node as the propositional formula which represents the set of documents which one would classify under the node, given that it has a certain label and that it is in a certain position in a tree. The tree mapping problem is reformulated into a set of node mapping problems. It identifies critical points in the mapping process, namely mapping elements with no relation where a stronger relation (e.g., generality) should have taken place, then attack critical points by exploiting sophisticated element level matchers which use the deep information encoded in WordNet, e.g., its internal structure, and taking into account the newly discovered information as additional axioms.

[7] uses the semantic web as background knowledge to show the feasibility and the potential advantages of using automatically selected online ontologies as background knowledge

for semantic mapping. It also enumerates major limitations of current ontology mapping approaches and the advantages of using the semantic web as background knowledge in detail. We refer the interested reader to it for a detailed discussion. Although this work uses the semantic web as a background knowledge well, but, like other approaches, it pays no attention to the role of context knowledge in the process of ontology mapping. In addition to introducing the role of context knowledge we extend the novel idea of this paper which is using the Semantic Web as background knowledge.

2 Using Context Knowledge As Structural Measure

Apart from lack of background knowledge, lack of context knowledge can affect the mapping result too. Saying that two concepts are the same, regardless what are the contexts they located in, is not rational. In this section we shed light on the importance of context knowledge in the process of mapping concepts of ontologies. We also try to propose a method to model it in order to use it for mapping ontologies.

We think of the exact meaning of a concept as not only a function of its label but also its context. In other words, the exact meaning of a concept is not only determined by the concept itself, but is also influenced by the surrounding concepts, and also by the situation in which the concept is used. To exemplify, consider concepts with hierarchical structures "$Images/B\&W/Europe$", "$Geography/Europe$" and "$Information/Images/B\&W/$-$Landscape$". The Europe in the first two hierarchical structures don't imply exactly the same concepts. The first one stands for black-and-white pictures of Europe while the latter one indicate geography information related to Europe. Hence these two concepts represent different concepts although they have exactly the same labels. The third concept mentioned above stands for black-and-white pictures of landscapes. Therefore, it is more similar to the first concept.

Having said that, we have to model the context knowledge in a proper way to improve the precision of our mappings. To achieve that, we present a novel method which extracts context knowledge from the ontology. We consider the set of all ancestors of a concept in the tree structure of the ontology as a key constituent of context knowledge. Then we define a measure as context similarity(CS) as the number of same elements of the two sets of contexts divided by the maximum number of them. Specifically, we define the context similarity (CS) for the two concepts C_1 and C_2 as follows:

$$A_1 = \{ancestors\ of\ C_1\}$$
$$A_2 = \{ancestors\ of\ C_2\}$$

$$CS = \frac{|A_1 \cap A_2|}{max(|A_1|, |A_2|)} \tag{1}$$

Thus CS is the proportion of common ancestors of the two concepts. Now, let's compute the (CS) measure regarding concepts mentioned above. The (CS) measure for the first and the second concepts is equal to 0 while it is equal to $2/3$ for the first and the third concepts. Roughly speaking, our hypothesis is that, the set of ancestors of a concept in it's hierarchical structure in the ontology is a good estimation of its context. After having computed the CS measure of the two concepts, we must consider it in the mapping algorithm not only for computing the similarity of the two concepts themselves but for their descendants. Hence, we try to find mappings in a top-down manner, from the root concept($Owl : Thing$) to the leaves in the hierarchical structures of ontologies.

It is important to note that a concept could be the child of two or more other concepts. In other words, a concept could be subclass of two or more classes. In such cases, we consider

the maximum similarity of such a concept with another concept as the CS value of the two concepts.

To sum up, we use the context knowledge to improve the precision of mapping algorithm even when the label of the concepts are the same.

Now, we define Structural Similarity(SS) of two concepts C_1 and C_2 as a function of similarity of their above concepts and their direct children as follows:

$$
\begin{aligned}
B_1 &= \{direct\ children\ of\ C_1\} \\
B_2 &= \{direct\ children\ of\ C_2\} \\
TS &= \frac{|B_1 \cap B_2|}{max(|B_1|, |B_2|)} \\
SS &= \frac{CS + TS}{2}
\end{aligned}
\tag{2}
$$

3 Using Background Knowledge

Methods that rely on a context ontology overcome two major limitations of traditional techniques. First, as observed by [Aleksovski et al., 2006], traditional methods fail when there is little lexical overlap between the labels of the ontology entities, or when the ontologies have weak or dissimilar structures. Indeed, these techniques are based on the hypothesis that syntactic correspondences imply semantic relations. While in many cases meaningful mappings can be derived from string and structural similarities, this hypothesis is far from being always verified. For instance, the relation between the concepts Beef and Food may not be discovered on the basis of syntactic considerations, but becomes obvious when considering the meaning of these concepts (their semantics). By ignoring such semantics, syntactic techniques fail to identify several important mappings. Relying on a context ontology allows deriving mappings in cases when the compared ontologies are dissimilar in their labels and structure. Indeed many alignments between dissimilar structures that cannot be found with traditional techniques are likely to already exist in the context ontology.

As alluded to, most state of the art systems, heavily rely on label and structure based similarity measures for mapping ontologies. As a consequent, on the contrary to what they claim, recent industrial-strength evaluations of mapping systems, see, e.g., [8, 9], show that they perform not with such high values of recall ($\sim 30\%$) as in cases of toy examples, where the recall was most often around 90%. Lack of background knowledge, most often domain specific knowledge, is known as one of the most prominent reasons for such low values of recall. The main contribution of this work is demonstrating the prominent role that background knowledge and context knowledge play in ontology mapping and presenting novel approaches to capture them in order to alleviate the side effects that may be produced on account of disregarding them in the process of mapping concepts of two ontologies.

There are multiple strategies to attack the problem of the lack of background knowledge. One of the most often used methods so far is to declare the missing axioms manually as a pre-match effort. Some other plausible strategies include:

- Extending stop word lists.
- Expanding acronyms.
- Reusing the previous match results.

- Using (if available) domain specific sources of knowledge.
- Querying the web.

Needless to say, the goal of ontology mapping is to build semantic bridges between ontologies using semantic relations that may exist between their elements. Nevertheless, all the aforementioned strategies to tackle the obstacle of lack of background knowledge depend only on textual and structural based similarity measures rather than semantic similarity measures. Hence, if there would be a great difference between the labels and structures of two semantically related concepts in two ontologies, they would fail to produce a promising result.

3.1 Using the semantic web as background knowledge

Using the semantic web as background knowledge has some advantages that the strategies mentioned above have not. The first one is that it doesn't need to select the source of background knowledge manually and it can be automatically selected. The second one is that, more or less, it covers all different domains, although, by now, it has rich and enough ontologies only for some domains, not for all domains. But with the rapid development of the semantic web it will have sufficient ontologies for every specific domain in the foreseeable future. Last, but not least, we can deduce different relations between a pair concepts, not only the similarity relation, and the relations between the concepts are the semantic relations that fit finely in the applications of the semantic web.

Here we use Swoogle to find semantic relations between pair concepts. Swoogle is a search engine for Semantic Web documents, terms and data found on the Web. It has indexed millions of Semantic Web Documents and employs a system of crawlers to discover RDF and OWL documents, rather than plain HTML documents. Swoogle provides services to human users through a browser interface and to software agents via web services (http://swoogle.umbc.edu/). Several techniques are used to rank query results inspired by the PageRank algorithm developed at Google but adapted to the semantics and use patterns found in semantic web documents.

In order to find relations between two concepts in the Swoogle, we search it to find ontologies that contain one of the concepts, say the first concept, and look for direct/indirect relations that may that concept have with the other one in the ontology. Considering the following general relations,

- C_1 rdfs:subClassOf C_2.
- C_1 owl:equivalentClass C_2.
- C_1 owl:sameAs C_2.
- C_1 owl:differentFrom C_2.
- C_1 owl:disjointWith C_2.

looking for relations between the two concepts in an ontology will come to an end in one of these situations:

1. There is a direct relation between the two concepts
2. There is an indirect relation between the two concepts

3. There is no direct/ indirect relation between the two concepts in the ontology

In the case where there is a direct relation between the two concepts, the simplest case, we can use that relation as the semantic relation for the two concepts but in order to improve the precision of inferred semantic relation we can continue our search for finding more rules involving the two concepts and finally infer the final semantic relation from them. In this situation we may encounter the problem of inconsistency between found relations that should be handled somehow.

In the case where there is an indirect relation between the two concepts in the ontology, we need an inference engine to deduce the best semantic relation between the two concepts. As with the above case the search process can be proceeded to find more relations (if available direct relations) between concepts and infer the final semantic relation from them.

In both the above situations we sacrifice precision for the sake of speed/ efficiency. Hence, in both cases we use the first found relation for deducing the semantic relation between concepts.

In the case where there is no direct/ indirect relation between the two concepts in the ontology we can use the relations that these two concepts may have with third concept and finally infer the semantic relation between the concepts from those relations. For example, having the following two relations,

- MasterThesis $rdfs : subClassOf$ StudentReport.
- StudentReport $rdfs : subClassOf$ TechnicalReport.

and using the transitive property of "$rdfs : subClassOf$" relation we can infer the following relation.

- MasterThesis $rdfs : subClassOf$ TechnicalReport.

If all the relations were transitive relations deducing the semantic relation between the concepts from any combination of them would be an easy and straightforward process. But because only the first three relations are transitive relations, the last two one are not, so inferring semantic relation from indirect relations of two concepts in an ontology is not so easy.

It is important to note that except the first relation mentioned above $(rdfs : subClassOf)$, other relations are used rarely in definitions of ontologies. Hence, we rely mainly on this relation.

In fact, using ontologies of the semantic web as background knowledge for matching concepts of ontologies we take into account the knowledge of users in ontology matching so it is tantamount to say that by using semantic web as background knowledge we make the user participation automatic.

4 Our Approach

We developed an application to test the applicability of using context knowledge and background knowledge in ontology matching and tested it with the OAEI test cases to evaluate it. Due to lack of space we avoid going into the details of the implementation and report here only what we have realized to be the most important findings.

Step 1: We first compare the concepts of the two ontologies by computing the textual similarity of them and match the concepts that their textual similarity is above 0.90 and have at least one third of their parents in common ($CS > 1/3$), for those concepts with hierarchical structure. We conducted some experiments to find the best threshold values. For example to

find the best textual similarity threshold we conducted some experiments and choose the point in which F-measure has the maximum value.

$$F\_Measure = \frac{2 * Presicion * Recall}{Presicion + Recall} \quad (3)$$

Step 2: After excluding the matched concepts of previous step from the list of concepts of

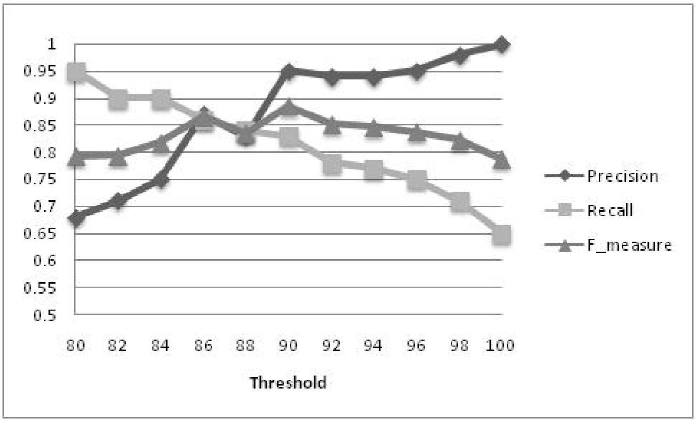

Figure 1: Find textual similarity threshold

the two ontologies, we continue our matching process by computing the WordNet similarity for the unmatched concepts. Computing the WordNet similarity of two concepts is somewhat time consuming, therefore we classify the concepts of each ontology in two groups (Classes and Properties) and compare each concept only with those concepts of the other ontology that are in the same group.

Here we use the wup (Wu and Palmer, 1994) similarity measures that is based on path lengths between a pair of concepts. It finds the depth of the LCS(least common subsumer of concepts A and B) of the concepts, and then scales that by the sum of the depths of the individual concepts. The depth of a concept is simply its distance to the root node. Setting the WordNet similarity threshold to 0.90 will produce good matching results. Analogous to step 1, for those concepts with hierarchical structure having at least one third of their parents in common($CS > 1/3$) is a prerequisite for matching them.

Step 3: In this step we compute the structure similarity(SS) measure for those class concepts with WordNet similarity above 0.70 and match those that their structural similarity are above 0.50. Unfortunately, because only class type concepts have hierarchy, therefore we can use this measure only for those concepts not for property concepts. Of course in some cases some of the properties have hierarchy structure too. But they are not so prevalent, hence unfortunately we cannot use structure similarity for them effectively.

Step 4: After completing the above three steps, the remaining unmatched concepts are those concepts that have little textual and lexical similarity with each other. Hence, we need an approach that goes beyond comparing concepts textually or lexically. Here is the point that our solution fits. Using the semantic web as background knowledge to discover hidden relations would be a good choice. Computing the semantic similarity of two concepts by searching in Swoogle is quite time consuming, even more time consuming than computing WordNet similarity. Hence, we compute the semantic similarity only for those concepts that have a high WordNet similarity, say above 0.70. It will significantly speed up the time the execution of the application takes, nevertheless, it has trivial affect on the results so that it

test case	falcon		foam		ola		our approach	
	precision	recall	precision	recall	precision	recall	precision	recall
201	1.0	0.95	0.96	0.90	0.92	0.88	1.0	1.0
205	0.88	0.87	0.89	0.73	0.43	0.42	0.92	0.76
208	0.97	0.95	0.94	0.82	0.87	0.64	0.99	0.88
209	0.88	0.87	0.86	0.86	0.78	0.58	0.91	0.72
230	0.94	1.0	0.94	1.0	0.95	0.97	1.0	0.94

Table 1: OAEI 2005 tests

test case	our approach without ...		our approach with ...	
	precision	recall	precision	recall
301	0.73	0.58	0.88	0.74
302	0.76	0.54	0.89	0.73
303	0.56	0.48	0.72	0.70
304	0.81	0.68	0.96	0.87

Table 2: testing the contribution of context knowledge and background knowledge in our approach

is negligible. It is simply because the WordNet similarity is a good approximation of the semantic similarity up to the point that some methods use it as a semantic measure rather than a lexical measure.

Having WordNet similarity above 0.70, we will match those concepts that have a relation of type $\{rdfs : subClassOf, owl : equivalentClass, owl : sameAs\}$ with each other either directly or indirectly and exclude those that have a relation of type $\{owl : differentFrom, owl : disjointWith\}$.

5 Evaluation

To test our approach we have used the Ontology Alignment Evaluation Initiative 2005 test suite [10]. The evaluation organizers provide a systematic benchmark test suite with pairs of ontologies to align as well as expected (human-based) results. The ontologies are described in OWL-DL and serialized in the RDF/XML format. The expected alignments are provided in a standard format expressed in RDF/XML and described in [10]. We compare the result of our approach with the results of three famous ontology matching systems. The test result is shown in table 1 for some test cases.

Rows correspond to test numbers, while columns correspond to the obtained values of precision (the number of correct alignments found divided by the total number of alignments found) and recall (the number of correct alignments found divided by the total of expected alignments). We improved the results especially the precision value by considering context knowledge and background knowledge. Having a richer and stronger reasoner our approach can outperform other systems. We conduct another test on four real-life ontologies of bibliographic references, ($3xx$ of test cases) that were found on the web and left mostly untouched, to investigate the effect of considering context knowledge and background knowledge in our approach and compute the precision and recall on those test cases with and without that factor. The results are shown in table 2. According to the values of the last two columns of the table 2 it is obvious that considering context knowledge and background knowledge have considerably improved the results in compare with the situation that disregards them.

In another experiment we tested the effect of context knowledge and background knowledge individually on precision and recall. Due to lack of space we avoid the details of this experiment but the most important finding that we achieved in this experiment is that context knowledge contributes more in precision and background knowledge in recall. Using both context knowledge and background knowledge in companion with each other we can achieve promising result as it shown in table 1.

6 Conclusions and Future Works

In this paper we discuss about the roles that context knowledge and background knowledge play in ontology matching process. We introduce a novel approach to capture the context knowledge and also discuss about advantages of using the Semantic Web as background knowledge in detail.

Although, the Semantic Web, as said before, by now, has rich and enough ontologies only for some domains and we use from only a limited set of existing relations, but the achieved results are so promising and if we have a rich and comprehensive set of ontologies it will be a very good source of background knowledge that can satisfy the needs of ontology matching well.

Future work includes a complete analysis of ontology structure and content and develop the solution so that it can use the existing relations between concepts in different ontologies as much as possible, and further development of the context knowledge capturing strategy.

References

[1] Noy Natalya. Semantic integration: A survey of ontology-based approaches. *ACM SIGMOD Record*, 33(4):65–70, 2004.

[2] Pavel Shvaiko and Jérôme Euzenat. A survey of schema-based matching approaches. pages 146–171, 2005.

[3] Zharko Aleksovski, Michel C. A. Klein, Warner ten Kate, and Frank van Harmelen. Matching unstructured vocabularies using a background ontology. In *EKAW*, pages 182–197, 2006.

[4] Zharko Aleksovski and Michel C. A. Klein. Ontology mapping using background knowledge. In *K-CAP*, pages 175–176, 2005.

[5] Zharko Aleksovski, W. ten Kate, and F. van Harmelen. Ontology matching using comprehensive ontology as background knowledge. In P. Shvaiko et al., editor, *Proceedings of the International Workshop on Ontology Matching at ISWC 2006*, pages 13–24. CEUR, 2006.

[6] F. Giunchiglia, P. Shvaiko, and M. Yatskevich. Discovering missing background knowledge in ontology matching. In *European Conference on Artificial Intelligence (ECAI)*, Trentino, Italy, August 2006.

[7] Marta Sabou, Mathieu d'Aquin, and Enrico Motta. Using the semantic web as background knowledge for ontology mapping. In *International Workshop on Ontology Matching (OM)*, Masters Hall, 1st floor, GA Center, Athens, Georgia, USA, November 5 2006.

[8] Jérôme Euzenat, Heiner Stuckenschmidt, and Mikalai Yatskevich. Introduction to the ontology alignment evaluation 2005. In *Integrating Ontologies*, 2005.

[9] Paolo Avesani, Fausto Giunchiglia, and Mikalai Yatskevich. A large scale taxonomy mapping evaluation. In *International Semantic Web Conference*, pages 67–81, 2005.

[10] Ontology alignment evaluation initiative. page http://oaei.inrialpes.fr/2005/, 2005.

Formal Ontologies Meet Industry
S. Borgo and L. Lesmo (Eds.)
IOS Press, 2008

Counterfeits and copies. An ontological analysis

Massimiliano Carrara [a,1] and Marzia Soavi [a]
[a] *Department of Philosophy – University of Padua (Italy)*

Abstract. *Counterfeits* and *copies* are central notions in e-commerce. But an adequate ontological characterization of them is not still available. This paper tries to fill the gap. It offers an ontological analysis of the notion of *counterfeit* as opposed to that of *copy*. While it seems clear that there cannot be a copy without an original, for instance, of a picture, it is far from clear that a counterfeit needs an original object in the same way. In the paper we analyze some proposals of what a counterfeit is: a first one (D1) closely connecting the notion of *counterfeit* and *copy*, a second one (D2) according to which a counterfeit presents itself falsely as having a certain origin that would give it more value than it actually has, and a third proposal (D3), elaborating the second, which takes into account the supposed origin, the type of the object in question, and the intent to mislead. We argue that none of the three proposals is without problems. Exploiting the notion of historical property we propose that being a *counterfeit* or a *forgery* is dependent on the intention of the producer (D4) and independent of the notion of *copy*, even if there are some similarities between the two notions.

Keywords. Artifacts, artworks, counterfeits, copies, ontology for e-commerce.

Introduction

In 2006 the French Union of Manufacturers (Unifab) accused eBay not to being proactive enough in coping with the problem of counterfeits sold through its website. eBay replied that it "is committed to protecting the intellectual property rights of third parties". Its sellers' policies confirm that "[c]ounterfeits, unauthorized replicas, unauthorized items (such as counterfeit watches, handbags, or other accessories) or unauthorized copies (such as copies of software programs, video games, music albums, movies, television programs, or photographs) are not permitted on eBay." [2]

Following eBay's sellers' policies, counterfeits and unauthorized copies are cases of unauthorized items. As the above eBay case demonstrates *counterfeits* and (unauthorized) *copies* are basic notions in e-commerce. But an adequate ontological characterization of them is not still available. This paper tries to fill the gap.

The aim of the paper is to offer an ontological analysis of the notion of *counterfeit* as opposed to that of *copy*. While it seems clear that there cannot be a copy without an original, for instance, of a picture, it is far from clear that a counterfeit *depends* on the existence of an original in the same way. In the paper we analyze some proposals for what a *counterfeit* is. We concentrate our analysis on *counterfeits* and *copies* of

[1] Corresponding author: Massimiliano Carrara, Department of Philosophy – University of Padua, P.zza Capitaniato 3, 35139 Padova, Italy; Email: massimiliano.carrara@unipd.it.
[2] See: www.theregister.co.uk/2006/08/22/french_counterfeit_crusade/

artworks, trying to generalize our definition to *artifacts*. Specifically, we analyze the following three definitions:

> x is a counterfeit of y if and only if (iff) x is a supposed copy of y and y is an original object. (D1)

> x is a counterfeit of y iff x is an object falsely purporting to have the history of production required for an original y. (D2)

> An object x is a forged XY – where 'X' is a term for a source of issue and 'Y' a term for a kind of thing forged – iff it is not a genuine XY but it is represented as a genuine XY with the intention to deceive. (D3)

In paragraph (§2) and (§3) we analyze and criticize the three definitions mentioned above. We show that even if (D1) – (D3) explicitly or implicitly take into account the notion of *copy*, none of them sheds light in the correct way on the relation between *counterfeit* and *copy*. In (§4) we isolate some relevant properties of *copies* and we sketch a characterization of this notion. In paragraph (§5) we propose our characterization of *counterfeit* based on a different relation between copies and counterfeits. We argue that:

> x is a counterfeit iff x has been intentionally produced to convince someone else that x has an *historical property* that it actually does not possess. (D4)

Unfortunately, even our definition is not without problems. Some difficulties of (D4) are sketched in the conclusions (§6) of the paper.

1. On a simple relation between counterfeits and copies: a first definition of counterfeits (D1)

It is very usual to associate the terms "counterfeit" and "copy". Consider the following examples taken from two English and one Italian dictionary. The *Webster's English Dictionary* definition of "counterfeit" is: "A copy that is presented as the original"; for the *Webster's Third New International Dictionary* a counterfeit is: "something that is made or produced as a copy: an artificial likeness". An Italian dictionary, the *Vocabolario della lingua italiana* of the *Istituto dell'Enciclopedia Italiana*, assigns also the following meaning to "copia": "disegno, pittura, scultura che riproduce più o meno fedelmente, a scopo di contraffazione o anche per esercitazione, per la diffusione, un'opera d'arte originale".[3]

From the above characterizations we could preliminarily conclude that copies made with the intent to deceive are said to be *counterfeits* or *forgeries.* Counterfeits and copies share the property of being reproductions of an original object. Such a characterization would lead us to give a negative answer to a question like:

[3] The English translation of the Italian entry is: "Drawing, painting, sculpture that more or less accurately reproduces an original artwork, with the aim of forging or for the sake of practice, or to spread (knowledge of it)"

Is it possible that all paintings are counterfeits or that all coins are counterfeits?

For, if every painting or every coin were counterfeit, there would be no original paintings or coins, and the questions: "Counterfeits of what?", "Forgeries of what?" would be without an answer. An answer to the above questions will be precluded in principle if it were possible that all coins or all paintings were counterfeits, and that no coin and no painting would be an original. So, there must be at least one coin or one painting of which counterfeits were copies.

If we accept this commonsensical way of relating copies and counterfeits (or forgeries), one could preliminarily conclude that there is an *ontological dependence* of counterfeits on original objects. One way to give an account of this dependence is to argue that it could not be the case that *all* paintings or *all* coins were counterfeits. Consider the following two arguments (A) and (B):

(A)

Some coins are sometimes forged. (PA)

Therefore, it could be the case that all coins are forged. (CA)

(B)

Some paintings are sometimes forged. (PB)

Therefore, it could be the case that all paintings are forged. (CB)

One could easy argue against the validity of (A) and (B) in the following way (see [4], [5]). Regarding (A): take a state where it is impossible to falsify coins just because there is no state mint. In that state, the counterfeiter would not know how to falsify a coin. There could not be fake coins just because there are no coins produced with the right material, by the appointed authority, coins of which the fake coins are counterfeits. So, it is impossible that coins are always counterfeits. If the above conclusion is right, (A) is an invalid argument. Another train of thought is at work against (B). A forged painting is a copy of an original one. It could not be the case that all paintings were counterfeit, for if there is no original painting there is nothing of which the counterfeit paintings are forgeries. So, there must be at least one painting of which paintings are *copies*.

Following the above arguments against the validity of (A) and (B) it has been argued that:

x is a counterfeit of y iff x is a supposed copy of y and y is an original object. (D1)

The problem of (D1) is that a counterfeit painting is not always a copy of a certain original object. Take, for example, the notorious case of the forged Vermeer made by

Van Meegeren: Van Meegeren's forged Vermeers were not copies of some original painting made by Veermer.[4]

This example does not demonstrate that it is possible that all paintings are forgeries, but simply that there are no good reasons to deny it on the basis of the notions of *copy* and *counterfeits*. However, one could argue even for the strongest thesis that:

It is possible that *all* paintings are counterfeits or forgeries. (T)

Consider the following mental experiment. Let us imagine a world without paintings. There are just sculptors. An artistic genius, Gino, one day announced to the world that the next day he would reveal a new form of art: painting. After a brief description of the nature of this new form of art he locks himself in his studio to execute the first painting. During the time spent in the studio he is kidnapped, and a second artist, Lino, executes the first painting, making sure to attribute it to Gino. The new painting exists and it is a forgery. Hence, (T) – i.e. the thesis that it is possible that *all* paintings are forgeries – turns out to be true. If the argument is sound then: "not only does a forged painting not have to be a copy of a painting that exists (or did exist), it doesn't have to be a copy of, or in style of any painting at all. *Copy of* is not one of the analytical elements of *forgery*" ([9], 148).

Even if one does not accept the above strong conclusion – refusing (T) – he could deny (D1) observing that it is possible to argue that a *conceptual dependence* of the notion of *counterfeit* on that of *original object* does not imply an *ontological dependence* of the first object on the second. People arguing for the first case of *dependence* acknowledge that, even if it is true that in order to understand what a *counterfeit* is we must refer to the notion of *original object*, that does not imply that to have a counterfeit (or a forgery) there must be an original object. Hence, even if the conceptual dependence holds – and it is true that there are forgeries which are copies and it is also true that for every copy there must be an original object – it is still possible that there could be forgeries without an original object. Then it is false that every counterfeit depends on an original object, and (D1) is false.

2. Two alternative definitions of counterfeits: (D2) and (D3)

Let us analyze two definitions of *counterfeit* not involving, at least *prima facie,* the notion of *copy*. Consider the following definition of counterfeit of an artwork: "an object falsely purporting to have the history of production requisite for the (or an) original work of art"([1], 122).[5] According to the above characterization a first, alternative, definition of counterfeit is:

x is a counterfeit of y iff x is an object falsely purporting to have the history of production required for an original y. (D2)

[4] Hans van Meegeren was a Dutch painter. He was born in 1889 and he is considered one of the most ingenious art forgers of the 20th century. After many negative comments – his works were judged to be tired and derivative – he decided to prove his talent by forging paintings of some of the most famous artists, including Johannes Vermeer. He replicated so well the style and colours of the artist that the best critics and experts of the time regarded his paintings as genuine Vermeers.

[5] With "history of production" the author refers to any production item giving authenticity i.e. how or by whom the object is produced.

where y is a case of *autographic art*. In Goodman's words "a work of art is autographic if and only if the distinction between original and forgery of it is significant; or better, if and only if even the most exact duplication of it does not thereby count as genuine" ([1], 113). A painting such as, for example, a Rembrandt *Self-Portrait* is a specific item connected, historically, to the artist who produced it. It is a case of *autographic art*. By contrast, music, dance, theater, literature, architecture are *allographic arts*. The reason is that they are independent of the history of production. You can listen to a performance of Beethoven's Third Symphony even if it is performed from a contemporary print of the score. Granting this distinction, we can talk of counterfeits or forgeries only for the case of autographic arts.

A third, more general, definition of *counterfeit* adds to the requirement of the history of production in (D2) a further condition concerning the type of object in question and the intent to mislead:

> An object x is a forged XY iff it is not a genuine XY but it is represented as a genuine XY with the intention to deceive. (D3)

Where XY is composed of two schematic letters: 'X' designates a source of issue, while 'Y' designates the type of the forged object (see [8], [9]).

Typically a source of issue X is the artist who created the object, as for example P. Picasso or A. Warhol, but it is also possible to forge objects created by groups or in a particular artistic period. Generally speaking, everything concerning the origin of an object relevant for its authenticity can be a *source of issue*.

On the other hand, a name of a painting as "Guernica", or a name of a particular artistic period as "the blue period", or again the name of a particular artistic movement as "cubism" can substitute 'Y'.

People adopting (D3) have in mind a very wide notion of counterfeit or forgery according to which the *status* of a counterfeit or forgery is determined more by the *use* made of the forged object than by its properties: counterfeits or forgeries are objects presented with the aim of deceiving someone. Forgers are those who present an object as an object originating in certain circumstances (author, period etc.) with the aim of convincing someone, even if the object does not have the supposed origin.

Unfortunately (D3) runs into many difficulties. First of all, it is not at all clear what exactly a *source of issue X* is supposed to be in (D3). We can try to work out what people using it mean by considering the following claims: "Since, by definition, every artifact has a source of issue – a human source of issue – and everything with a source of issue is, in the relevant sense, an artifact, it follows that every artifact is logically capable of forgery, and everything logically capable of forgery is an artifact." ([9], 153). If we accept the most common notion of *artifact*, according to which an artifact is a material object intentionally produced by human beings – what makes 'artifactuality' dependent on the condition of origin – it is clear that "source of issue" refers only to conditions concerning the physical origin of the object. A counterfeit or a forgery is an object that is misleadingly represented as an object originating in conditions different from the real ones. According to (D3) the authenticity of an object must be assessed only with respect to the conditions concerning the origin of the object. Nonetheless, there seem to be cases where it is perfectly correct to distinguish between an original object and a fake one without taking into account their origin. Consider, for example, Alexander's armour. It is clear that the authenticity of such an object does not concern its origin: it will be *the* true Alexander's armour if it belonged to him or if Alexander

used it. Indeed, knowledge about origin conditions may be of some help for assessing its authenticity: if we know that some putative Alexander's armour dates to a period later than Alexander's death we can surely say that it is not an original. However, its origin does not ground its authenticity. So, we can distinguish between the property of *being original* – something always related to the conditions of the origin of an object – and the property of *being authentic*, a property that can also be related to different historical conditions. The point we want to stress is that, in order to devise a theory of counterfeits of artworks or, more generally, of counterfeits of artifacts, we need to distinguish between the originality and the authenticity of an object, and to take this second property as the one that is truly complementary to that of counterfeit or forgery. In the case of artworks we speak of a forged copy only with respect to some conditions concerning its origin; while for other artifacts originality can be irrelevant for their authenticity.

The second problem of (D3) concerns the other schematic letter *Y*. We said that 'Y' can be substituted by any term that designates a kind of artifact, or by the name of an object that is an artifact. It cannot be restricted only to artworks: even artifacts as, for example, bags and money could be forged. The problem is that it seems possible to speak of counterfeits or forgeries even for natural objects, not just artifacts. Consider, for example, the stone(s) used by David to kill Goliath or some fossils discovered on the surface of Mars. Even in these cases it does not seem senseless to wonder about their authenticity. Take, for instance, the case of fossils: the relevant conditions concern their origins; but fossils are not artifacts, hence the relevant conditions must be of a different type with respect to the origin of artifacts.

Finally, a third flaw in (D3) is due to the idea that an object is a forgery only if it is represented as an authentic *Y*. But, the representation must be a conscious misleading representation with the aim of deceiving. According to supporters of (D3), counterfeits or forgeries are distinct from a simple error of attribution because their representation is consciously misleading. For example, a critic who *bona fide* attributes a Giotto fresco painting to Cimabue is not generating a counterfeit or a forgery: he is simply making a mistake. Moreover, it seems very hard to speak of counterfeits or forgeries without referring to those who are to be deceived. Considerations concerning the competence of the agents to be deceived seem to be crucial for the forger's epistemic possibility of representing a certain object as a fake *Y*.

As we said earlier, (D2) and (D3) are very similar. Specifically:

A) both are historical definitions, i.e. they both involve a reference to the origin of the counterfeits or forgeries, and
B) both deny that being a copy is a necessary condition for being a forgery.

The main difficulties of (D2) are:

1. it cannot be used to tell apart forgeries and simple errors of attribution;
2. it is formulated making reference to the obscure notion of *purporting*;
3. it does not specify which are the properties of the origin relevant for the *status* of being a counterfeit or a forgery;
4. it defines the notion of *counterfeit* or *forgery* only for artworks.

The same kind of criticisms, with the exception of (1), could be applied also to (D3). In particular, we have already observed that:

5. it is formulated making reference to the obscure notion of *representing*;
6. it does not specify the relevant properties of the origin;
7. it is not true that it can be successfully applied to objects that are not artworks.

Moreover, it is not clear enough how it is possible to accept (B) denying (A): if an original object is necessary for defining counterfeits how is it possible that there should be no relation between counterfeits and copies? *Prima facie*, there seem to be common properties *counterfeits* and *copies* share, e.g. *being similar to an original object*. In the following paragraph we propose our analysis of the notion of *copy*. On the basis of this analysis we sketch our characterization of *counterfeit* or *forgery* (D4). We show that (D4) avoids the problems (1) – (7) that afflict (D2) and (D3).

3. Copies: some relevant properties and a definition (C8)

A copy is an artifact produced with the intention to reproduce some relevant features of an object taken as a model, called *original object*, or, in other terms:

If x is a copy of y, x is the product of human contrivance (C1)

and

If x is a copy of y, y is an *original object*. (C2)

From (C1) and (C2) follows that the relation of copy can hold only between distinct objects: we cannot say that a painting is a copy of itself in the same sense in which we can say that a good counterfeit or forgery is a copy of the original painting. At the same time we would never say that a painting is a copy of one of its forgeries as the forgery is a copy of the painting itself. So we can say that the relation we are discussing is irreflexive and asymmetric, i.e.:

$$\forall x\, (\neg\, (x \text{ Copy } x)) \tag{C3}$$

and

$$\forall x\, \forall y\, ((x \text{ Copy } y) \rightarrow \neg\, (y \text{ Copy } x)) \tag{C4}$$

Transitivity is less straightforward. If we define the copy as an artifact produced with the intention of reproducing some of the features of an object taken as a model – the original object – we cannot admit transitivity. For if x is modeled on y and y is modeled on z, it is not true that x is modeled on z itself:

$$\forall x\, \forall y\, \forall z\, ((x \text{ Copy } y)\, \&\, (y \text{ Copy } z) \rightarrow \neg\, (x \text{ Copy } z)) \tag{C5}$$

The reason for (C5) is that, given the intentional aspect of a copy, it is clear that the relation of copy cannot be transitive if by “the object taken as model” we refer to the object observed by the author during the process of copying.

Nevertheless, there seem to be some problematic cases, not involving the agent’s direct observation during the copying process. Take, for example, those cases in which an object x is modeled on a memory image of another object y. Even if the object is similar enough to the original one, on the basis of our rough definition, we cannot consider it as a copy. Nonetheless, usually we would consider x – the object modeled on a memory image – a copy of y. This observation could lead us to think that the expression “the object taken as model” has to be intended to refer not to an object observed during the copying process, but simply to the object that the author wants to imitate, and this raises a new problem.

Consider the case of a statue x that has been modeled on a copy of some ancient one, that is now lost; the intention of the author is to imitate the lost statue, knowing that a certain copy z of the original one is a good copy. In this case we can say that x is a copy of the lost original, but would not we say also that it is a copy of z, the statue observed during the copying process? We think that the answer to the above question is: yes!

So x would have two different original objects, and transitivity would hold. If we want to take into account also such cases it would be better to speak of *non-transitivity*, instead of *intransitivity*, of the copy relation:

$$\neg \forall x\ \forall y\ \forall z\ ((x \text{ Copy } y)\ \&\ (y \text{ Copy } z) \rightarrow (x \text{ Copy } z)) \qquad \text{(C6)}$$

According to the intentional notion of *copy* we commonly take to be the model of the copy just the object observed during the copying process. We are perfectly aware that the notion of *observed object* is highly problematic – that’s why cases like that of the memory images mentioned above are problematic. Adopting (C3), (C4) and (C5) it is possible to avoid transitivity, but indeed this fact involves a restriction with respect to the commonsensical notion of *copy*. For example, if we sculpt a new statue x just observing some photos of a statue y, x – according to our notion of *copy* – cannot be considered a copy of y never mind how much x resembles y. This choice is conditioned by the fact that we are considering just copies of material objects. Obviously, things seem to go differently for very common cases of copies like computer programs, files, video games, music albums, movies, where transitivity seems to be straightforward.

Apart from these cases, a non-transitive copy relation avoids the problem of a Sorites paradox for copies[6]. The reason is that if we allow for transitivity, it would be possible to create chains of copies of an original object, that are less and less similar to the original object, until we reach a *final one* which is far too different from the model to be considered a copy of it. In the case of this intransitive relation the Sorites paradox does not arise because copies are just those directly derived from the original.

These considerations introduce another basic feature of the *commonsensical* notion of a copy. An object x can be a copy of another object y only if it is similar enough to it. It seems to be quite intuitive that:

---

[6] The general form of a Sorites paradox can be easily understood looking to one of its most famous formulations: Premise 1) someone with 0 hair is bald; Premise 2) if a person with n hair is bald a person with n+1 hair is still bald; Conclusion) a person with 10,000 hair is bald. The paradox is due to the fact that even if the premises are all true, the conclusion is patently false.

If x is a copy of y, x resembles y. (C7)

Indeed, the vagueness of *resemble* gives rise to some obvious difficulties. There are degrees of resemblance and most of the time the resemblance is evaluated in relation only to some properties of the objects. Commonly, we accept to consider an object as a copy even if it is not indistinguishable from its original: we can talk of better and worse copies, and even in the case of good copies, it does not seem to be necessary that they be absolutely indistinguishable – i.e. indistinguishable with regards to all the relevant properties – from their original objects.

A *commonsensical* notion of *copy* seems to waver between two aspects: one dealing with the *physical properties* of the object, the other with the *intentional ones*. Examples of physical properties are *colors*, *shape* and *weight*, while *being intentionally produced* is an example of what we call *intentional property*. From an intentional point of view, something is a copy if it has been produced with the intention of producing an object similar enough to the one taken as a model, while from a physical point of view something is a copy if it is physically similar enough to its original object. This second aspect is clearly dependent on the first one since it seems possible to say which object is the designated original one only with reference to the intentions of the author.

Moreover, we can recognize an ideal notion of *copy* where a copy and its original are absolutely indistinguishable, except for their spatial location. Such copies are very difficult to realize, and we call *copies* only those objects that are similar enough to their originals. This aspect of the notion of *copy* is revealed also by the use of expressions like “to be a good copy” or “this one is a better copy than that one”. For this reason, it seems to us that the *status* of copy concerns two different aspects of the object, the physical and the intentional.

On one hand, it is necessary for a copy to have been produced according to an intention of a certain type; on the other hand, a copy has to be similar enough to its original one. The physical constraint concerns the minimal similarity that an object must have to be considered a copy. Both of these conditions have to be satisfied for the object to be a copy. And we regard them as jointly sufficient.

Hence, our proposal is that what is relevant to something’s being a *counterfeit* is a notion of *copy* including both *intentional* and *physical* properties, related to similarity:

x is a copy of y iff x has been produced with the intention of making something similar to y and x actually resembles y. [7] (C8)

4. A new definition of *counterfeit* (D4)

We maintain that to be a *counterfeit* or *forgery* depends on the intentional conditions connected to the *origin* of a certain object. Following this line of thought we propose the following definition of counterfeit:

x is a counterfeit iff x has been intentionally produced to convince someone else that x has an *historical property* that it actually does not possess. (D4)

We think that (D4) avoids the main problems of (D2) and (D3), i.e. for (D2) that:

[7] On a different notion of copy see [6], [7].

1. it cannot be used to tell apart forgeries and simple error of attribution;
2. it is formulated making reference to the obscure notion of *purporting*;
3. it does not specify which are the properties of the origin relevant for the *status* of being a counterfeit or a forgery;
4. it defines the notion of *counterfeit* or *forgery* only for artworks.

And for (D3) that:

5. it is formulated making reference to the obscure notion of *representing*;
6. it does not specify the relevant properties of the origin;
7. it is not true that it can be successfully applied to objects that are not artworks.

The first advantage of (D4) over (D2) and (D3) is related to (1). In fact, the problem of the distinction between *counterfeit* and *misattribution* seems to fade away for (D4). The problem arises in (D2), because a counterfeit is an object represented or used in a certain way by the forger. In (D4) we state the dependence of *counterfeits* on the intention of the producer, and not on the representation of the user. It is not possible to confuse counterfeits and misattributions: counterfeits have to do with the intentions of the producer, misattributions with estimations made by some agents.

The second advantage of (D4) over (D2) and (D3) is related to (2) and (5), because (D4) avoids the obscure use of the notions of *purporting* and *representing*. There is no mention of them in (D4). However, one could reply that some other problematic notions – like that of *intentional production* are adopted in our formulation (D4). Let us observe that the most problematic aspect of the notion of *intentional production* concerns intentionality, and intentionality is implicitly involved also in the notions of *purporting* and *representing*. Moreover, even (D2) and (D3) deal implicitly with the notion of *intentional production*. Hence, on this point, our solution comes off better than (D2) and (D3), because, at least, avoids the use of the above problematic notions.

The third and main advantage of (D4) is that it can cover those *problematic* cases related with the specification of the origin conditions – (3) for (D2) and (6) for (D3). In (D4) we specify the relevant conditions of the origin as a certain kind of *historical property* of the original object faked by the producer of the counterfeit. A *historical property* is a property that something possesses merely in virtue of having been involved in some past event.

A further advantage of (D4) is that the intention of the producer may be that of trying to fake not only historical properties related to the origin of the authentic object but also historical properties related to its history. Hence, (D4) allows for an account of the fact that it seems plausible to speak of counterfeits of an object belonging to a historical or famous personage – as for example Alexander or Paris Hilton – or coming from a certain place, for example Mars, or simply used in some specific circumstances, like David's stone.

As regards (D1) we agree with the criticisms mentioned above, and reject that counterfeits need to be copies of an original object. However, we recognize that there is a certain relation between *counterfeits* and *copies*. There are at least two commonalities between them, as defined in (D4) and (C8):

C) For *counterfeits* and *copies* the intention related to their production has a key role;

D) For *counterfeits* and *copies* it is necessary that the copy and the counterfeit be *sufficiently* similar to the object taken as a model. In fact, even for counterfeits some level of similarity must be granted in order to convince someone.

Nonetheless C) and D) can at most be exploited to show that being a copy and being a counterfeit are similar notions, but in our view being a copy it is neither necessary nor sufficient for being a counterfeit. There are kinds of counterfeits, that Goodman first recognized and named “creative counterfeits” for which it is not the case that they are copies of some original object (see on this [2], [3]). A counterfeit of this kind would be, for example, Van Meegeren’s forged Vermeers. On the other hand, for counterfeits of single objects, being a copy is not sufficient for being a counterfeit, at least if we adopt the (C8) notion of *copy*. I can make a perfect copy of the Guernica just for practice and this does not make me a forger.

We agree with (D3) that in speaking of counterfeits we commonly presuppose that a certain attempted act of cheating about the origin of the object has taken place. Nonetheless, it is extremely important to distinguish the level of actions and that of objects. It is plausible, for example, that an object that was not meant to be a forgery might be used as a forgery, but to be a forgery and to be used as a forgery, we claim, are two different properties.

5. Some concluding remarks on (D4)

(D4) is not without its own problems. Consider the following, problematic cases. First case: Pino makes a copy of a Picasso for a study; Pina steals Pino’s painting; and unbeknownst to Pino she sells it on eBay as an original Picasso. In this story, Pino does not produce a painting to convince anyone that it is an original Picasso, hence according to (D4) the painting is not a counterfeit. This is, for sure, a flaw of our definition.

Obviously, we can say that Pino’s painting has been used *as a counterfeit*. In other words “counterfeit” may be considered as an ambiguous term: some of its uses refer to objects, some others to actions.

Second story: Pino makes a copy *a* of a Picasso for a study; he recognizes himself as a very talented painter; he decides to start production of fake painting, counterfeits, and to sell them, via eBay; he produces a second copy *b* of another Picasso painting and then he sells both *a* and *b* as authentic Picassos, via eBay. In line with (D4) we cannot say that *a* is a counterfeit while we can say that *b* is a counterfeit, and, of course, this is a very counterintuitive consequence.

Even in this second scenario it appears that “counterfeit” refers to a certain use of an object i.e. in virtue of the mere fact that an object is passed off as an original, it can be considered a counterfeit.

Acknowledgments

We would like to thank Richard Davies, Gabriele Tomasi and the referees of FOMI08 for their detailed comments to earlier versions of this paper.

References

[1] N. Goodman, *The Languages of Art. An Approach to a Theory of Symbols*, Hackett Publishing Company Inc., Indianapolis (Indiana), 1968.
[2] N. Goodman, A Note on Copy, *The Journal of Aesthetics and Art Criticism* **44** (1986), 291-292.
[3] J. Levinson, Autographic and Allographic Art Revisited, *Philosophical Studies* **38** (1980) 367-383.
[4] G. Ryle, *Dilemmas*, Cambridge University Press, Cambridge, 1954.
[5] J. Rosemberg, *The Practice of Philosophy,* Prentice-Hall Inc., Englewood Cliffs (NJ), 1978.
[6] A. Tzouvaras, A., Significant Parts and Identity of Artefacts, *Notre Dame journal of Formal Logic* **34** (1993), 445-452.
[7] A. Tzouvaras, A World of Homogeneous Artefacts, *Notre Dame Journal of Formal Logic* **36** (1995), 454-474.
[8] M. Wreen, Goodman on Forgery, *The Philosophical Quarterly* **133** (1983), 340-353.
[9] M. Wreen, Forgery, *Canadian Journal of Philosophy* **32** (2002), 143-166.

Application Scenarios of Ontology-Driven Situation Awareness Systems

Exemplified for the Road Traffic Management Domain

Norbert BAUMGARTNER [a,1], Werner RETSCHITZEGGER [b] and Wieland SCHWINGER [b]
[a] *team Communication Technology Management Ltd., Vienna, Austria*
[b] *Johannes Kepler University, Linz, Austria*

Abstract. Large-scale control systems, as encountered in the domain of road traffic management, typically deal with highly-dynamic environments providing information about a large number of real-world objects, which stem from multiple heterogeneous sources and are anchored in time and space. Human operators of such systems face information overload which endangers the recognition of critical situations. Situation awareness systems should support operators fulfilling their tasks by leveraging their awareness of the ongoing situations. However, current approaches to SAW miss a common conceptual model necessary for various aspects of SAW. Although the application of ontologies for filling this gap has been proposed in recent years, ontology-driven SAW systems are nevertheless still in their infancy. In this paper, we shape the vision of an ontology-driven SAW system by the analysis of application scenarios facilitating the features of formal ontologies. We illustrate the suggested scenarios with examples from the field of road traffic management and argue that an ontology-driven SAW system does not replace but may actually enhance traditional probabilistic approaches to SAW.

Keywords. Ontologies, Situation Awareness, Context Awareness, Road Traffic Management

Introduction

Large-scale control systems, as, for example, in use in the domains of road traffic management or air traffic control, operate in geographically wide-spread environments and involve - partly incomplete - information about mainly physical objects (e.g. traffic jams, accidents, roadworks) from heterogeneous information sources. Human operators of such systems face an increasing amount of information to be incorporated in order to timely and correctly resolve or even prevent critical situations. *Situation awareness* (SAW) applications support human operators by pinpointing their attention to these situations. In the field of situation awareness, a situation is usually defined as a set of inter-

[1]Corresponding Author: Norbert Baumgartner, team Communication Technology Management Ltd., Goethegasse 3/3, 1010 Vienna, Austria; E-mail: norbert.baumgartner@te-am.net.

related physical objects, i.e., situations aggregate information resulting in a decrease of information overload. Because this abstraction also entails a shift from numeric to rather symbolic information, ontologies have been proposed to provide the missing conceptual model of SAW in recent work (e.g., [1]). In this paper, we contribute an analysis of potential high-level application scenarios of ontologies for SAW toward an ontology-driven SAW system and briefly discuss their integration with traditional approaches to situation awareness. The application scenarios are illustrated with examples from the field of road traffic management.

The paper is structured as follows: After an introduction to our running example road traffic management in section 1, we provide an overview of situation awareness in general and present the corresponding state of the art in section 2. In section 3, we investigate application scenarios of ontology-driven SAW and provide an overview of current approaches including our work BeAware!, a framework for ontology-driven SAW. We conclude the paper in section 4 with a summary of our contribution and an overview of further prospects.

1. Road Traffic Management

In the field of road traffic management (RTM), the overall goals a road traffic operator has to achieve are the reduction of traffic jams and the prevention of accidents. The tools a traffic operator may resort to are direct control measures (e.g., the restriction of speed limits) or indirect control measures (e.g., via warning messages) [2]. In order to correctly take these measure in time, the traffic operator has to be aware of the ongoing or evolving traffic situations. However, with the recent advances in sensor technology and information systems, the information a traffic operator has to incorporate has dramatically increased—leading to the problem of information overload. Fig. 1 provides an illustrative example for the induced problems.

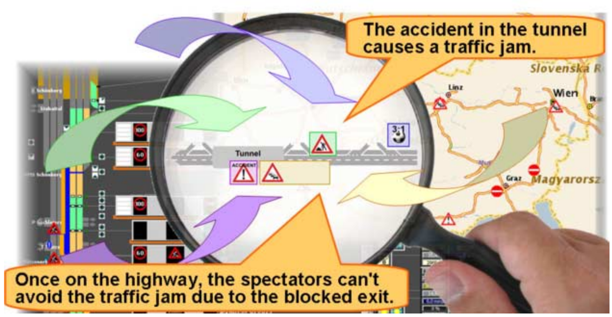

Figure 1. An illustrative example for a critical road traffic situation

The information, which have been arranged within the magnifying glass for our example, are ususally scattered across various graphical user interfaces and come from different heterogeneous information sources. Though the human operator may be aware of the accident in the tunnel and the resulting traffic jam, it may be difficult for him to realize that the exit blocked by roadworks would obstruct the soon leaving spectators of the football game from avoiding the traffic jam—cancelling the roadworks will be too

late. Though this example may be exaggerated it illustrates the dangers of missing SAW in RTM.

In the next section, we are going to have a more general look on these problems by introducing SAW and providing a brief overview of the state of the art in this area of research.

2. Situation Awareness

In this section, we have a look at the notion of SAW and its multiple origins in order to highlight its interdisciplinary significance. Subsequently, we provide an overview of current approaches and shortly state the gaps they leave for ontologies.

2.1. Different Views of a Single Problem

The original notion of SAW has been coined by Endsley in the field of human-computer interaction and cognitive sciences [3]. She stated that SAW is '*the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future*'. From this definition, Endsley derived three layers of SAW: perception, comprehension, and projection. Though information overload implies that the amount of perceptible information increases, comprehension and projection tend to be more difficult because of the large quantities of information. Endsley's definition has been embraced and refined by the area of information fusion, in which the process of gaining SAW in a computational way has been denoted *situation assessment*. Because of its ranking in the processing chain of a typical large-scale control system (cf. the JDL Data Fusion Model [1]), situation assessment is also referred to as *high-level* information fusion. The following example from the field of RTM should clarify this kind of layered separation of processing steps:

1. *Signal assessment*—on the lowest level, signals measured by sensors (e.g., magnetic fields) are interpreted as numeric features like the traffic flow or the average velocity.
2. *Object assessment*—physical objects (i.e. 'the elements in the environment within a volume of time and space' in Endsley's definition) are identified based on the numeric features of the lower level; e.g., a detected traffic jam or accident.
3. *Situation assessment*—by the derivation of relations between the identified objects, relevant situations are assessed, e.g. an accident that *causes* a traffic jam or a blocked exit that *obstructs* motorists. Note that the prior levels must not be present for each object, e.g. scheduled roadworks are also relevant for situation assessment, but are rather manually entered than automatically assessed.
4. *Impact assessment*—once the traffic operator is aware of the relevant situations, the impact can be deduced in order to select the actions to be taken, e.g. cancel roadworks.

Apart from the fields of cognitives sciences and information fusion, also the field of pervasive computing has introduced a notion which is similar to SAW: Context awareness. Though there are striking similarities, research in both areas is quite separated. One of the most significant differences is that SAW (at least in large-scale control systems) fo-

cuses on the human operator who *observes* relevant situations in contrast to the agent in the realm of context awareness which is typically *part of* the situations of interest. Thus, we rank SAW on a higher level of abstraction than context awareness.

Summarizing this subsection, the notion of SAW may be found in various research areas like cognitive sciences, information fusion, or pervasive computing, which highlights its interdisciplinary significance and which will provide the basis for motivating the application scenarios of ontology-driven SAW.

2.2. *State of the Art*

In this subsection, we provide a brief overview of the main approaches to SAW which are relevant for our work. Thus, we largely omit the field of cognitive sciences for the following discussion, since we are mainly interested in computational approaches to SAW[2].

In recent years, a shift of focus from the lower levels of information fusion to situation assessment has been observable. Whereas there is agreement that this shift implies the introduction of rather symbolic in contrast to the numeric information on lower levels (e.g., [4]), there is no agreed approach on how to actually handle these different requirements. Traditional approaches to SAW are of rather probabilistic nature and reach, for example, from the application of bayesian belief networks [5] to belief fusion by Dempster-Shafer models [6]. The common understanding is that SAW is not about objects, it is about relations, which is one of the most important differences from the lower levels of information fusion. A further common assumption is related with the focus of SAW system development—the success of a SAW system is determined during design time, e.g., during the analysis which traffic situations can occur, rather than on run time [7]. This assumption is supported by the fact that most related work reporting on the application of different situation assessment approaches rest on some conceptual model of the application domain—which is often taken for granted in related work. Moreover, a common abstraction of these conceptual models is missing or work in progress, which makes research results difficult to discuss (cf. [8], [4]). This is the moment when (formal) ontologies enter the stage. As promoted in related work (e.g. [1], [9]), ontologies could fill the important gap of providing a conceptual model for SAW.

Also in the area of pervasive computing the usage of ontologies has been proposed—for example, Strang et. al. [10] suggest the usage of ontologies due to their formality and the accompanying reasoning capabilities (especially in contrast to object-oriented models).

In the following section we point out for which SAW application scenarios we believe ontologies are beneficial. As mentioned above, we advocate the view that ontologies 'just' fill gaps and do not substitute traditional approaches. However, if one commits to an *ontology-driven* system as envisioned by Musen [11], the ontology becomes an integral part of of the system architecture. Thus, we also outline how to integrate traditional approaches to SAW in such architectures when presenting the application scenarios in the next section.

[2]Nevertheless, at least one application scenario of ontologies for SAW is going to be closely related to human-computer interaction.

3. Application Scenarios of Ontology-Driven Situation Awareness

The application scenarios presented in this section should point out the advantages of formalizing the conceptual model of a SAW system using an ontology. The application scenarios are illustrated by examples from the domain of RTM and are depicted at a glance in Figure 2. The assumed ontology definition language is OWL-DL[3] which is recommended by the W3C and should constitute the basis of the Semantic Web. The section is concluded by an overview of current approaches in the field of ontology-driven SAW.

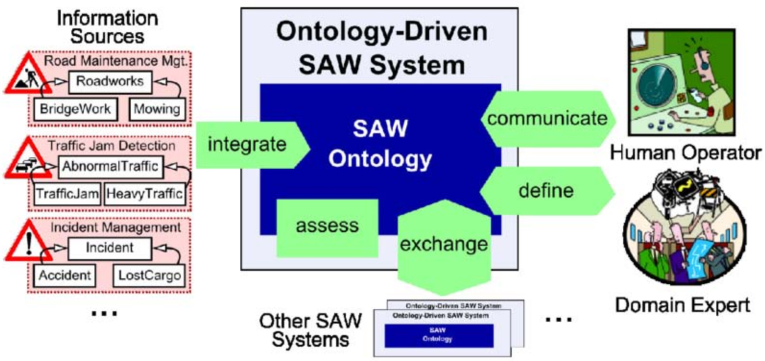

Figure 2. The application scenarios at a glance

3.1. Integrate Information Sources

One motivation for the usage of ontologies is that they may be used to integrate heterogeneous information sources (e.g., [12]). With the advent of the Semantic Web, the number of ontologies available for diverse domains is constantly increasing. The benefit of using such ontologies is that they provide an agreed specification of the concepts of a domain which can be used to describe the different information sources of a SAW system. Referring to our illustrative example from Sect. 1, each information source may be represented by a separate ontology (e.g. one for public events containing an individual representing our football game). Within the SAW system, these information sources could be mapped to a more abstract, domain-independent ontology that, as suggested in various related work (e.g., [13], [14]), fits the tasks of SAW. Note that this application scenario is especially important in the field of pervasive computing, in which it should be possible to integrate information sources on the fly.

3.2. Communicate with Human Operators

Another application scenario is to use ontologies for communicating the results of situation assessment to human operators. First of all, ontologies can provide human-understandable information about complex relations among objects. For example, an appropriate ontology may be in a position to express that 'an accident causes a traffic jam'

[3]Web Ontology Language, http://www.w3.org/TR/owl-features/

based on the assumption that 'accident' and 'traffic jam' are individuals of the corresponding classes and 'causes' is a property that has been derived during situation assessment. Moreover, current approaches to construct semantic (or even geographic) mashups of arbitrary information enable new ad hoc user interfaces (cf. [15]). Another related application scenario is the exploitation of the query mechanisms ontologies provide, which has already been analyzed for SAW systems by Kokar et. al. [9]. For example, a human operator may be interested in all traffic situations in which the visitors of a 'public event' are obstructed by a 'traffic jam'. Employing an appropriate ontology, 'football game' could be inferred to be a subclass of 'public event' and included in the answer of the query.

3.3. Exchange Knowledge About Situations

SAW systems are not isolated; for example, the Austrian highways agency regularly exchanges traffic information with its neighboring counterparts. Especially if traffic control strategies should be determined across jurisdictional boundaries, it is necessary to exchange all the available knowledge about occurring situations between potentially heterogeneous SAW systems. General advantages of ontologies enabling such a exchange are that they are machine-readable, platform-independent, and easily exchangeable via the Internet. Furthermore, an agreed domain-independent SAW ontology could provide the necessary common base for knowledge transfer between two systems. Nevertheless, the assessment of individuals, e.g., how the 'causes' relation between an 'accident' and a 'traffic jam' is derived, has to be system-dependent, which is, however, obvious—think of a traffic jam in an urban area and a traffic jam on a highway, both have completely different characteristics from a traffic engineer's point of view.

The application scenarios, which have been described so far, not necessarily induce a system-wide ontology-based conceptual model. Though tediously to implement, they may just represent the interfaces of a SAW system to information sources, the human operator, and other SAW systems. The last two, not less important application scenarios require, however, the SAW system to be completely ontology-driven. Thus, also the integration with traditional approaches to SAW as indicated in the previous section is discussed.

3.4. Define Situation Types

Following the notion introduced by Barwise and Perry [16], we view a situation type as an abstract state of affairs that may be instantiated during situation assessment. As already mentioned, we believe that the focus of successful SAW has to reside on the design time of a SAW system which particularly involves the definition of interesting situation types. Domain experts who have the task of designing a SAW system should be given a working language to define the characteristics of situation types. In fact, we are speaking of the common conceptual model which is missing in current approaches to SAW. A domain-independent ontology implementing this conceptual model for SAW could enable domain experts share their views as well as experiences across SAW systems and even domains, thereby laying the cornerstone for successful system behaviour at run time. However, such an ontology must not be reduced to a mere vocabulary. The value of this application scenario rather depends on the possibility to formally define the

constraints of a SAW system (e.g. the situation types of interest) and check whether the design is consistent with the agreed conceptual model the ontology provides. Regarding such a formalization using Semantic Web ontologies, we have to use a rule language on top of OWL-DL, because its instance classification features do not suffice for describing situation types. Examples are the application of the SWRL[4] as discussed in [17] or the a logic-programming-like rule engine as provided by the Jena Semantic Web Framework[5] which we applied in our previous work [18].

Whereas we have a look at such existing approaches below, we proceed our discussion about the definition of situation types by inspecting the, as already mentioned, most important aspect of SAW: Relations. Let us revisit the illustrative example from Sect. 1 and assume that traffic engineers have the task to define a situation type which captures the essence of the situation the human operator has been unaware of. An immediate question is how to define that an accident 'causes' a traffic jam. Of course, one could provide some proprietary interpretation of this relation, but it would be difficult to discuss the situation type without a common understanding of 'causes'. As suggested in our previous work (cf. [19], [20]), we believe that a common SAW ontology should incorporate spatio-temporal *primitive relations* like 'proper part', 'before', etc. which could provide the basis for defining relations like 'causes', 'obstructs', and so on. Although the concrete interpretation of these primitive relations would also be dependent on the domain or even on the SAW system, they are easier to understand and straight-forward to implement.

The consequence of such an application scenario, i.e. the definition of situation types based on a SAW ontology, implies that the ontology must be consulted for situation assessment turning it into an integral part of the a SAW system. We argue that the effect on the applicability of traditional approaches is negligible. Traditional probabilistic approaches could, for example, be used for refining and learning situation types. Given that our common SAW ontology contains relations and situations, they could also be regarded as nodes in a bayesian belief network. Thereby, we may define the relations a situation depends on based on the definition of the situation type. Based on the results of situation assessment, the network could incorporate actual evidence of co-occurring relations respectively situations and even find new structures (e.g. an unnoticed dependency between a relation and a situation). These results could again be incorporated into the definition of situation types.

3.5. Assess Situations

As indicated above, situation assessment is about pattern matching—in our case, the pattern is specified as a rule which should be instantiated by individuals in the common SAW ontology. The question is how to exploit the ontology for effective situation assessment. In fact, the a priori knowledge encoded in an ontology can be used to perform some efficient reasoning steps. Some examples, which we have introduced in previous work (cf. [20], [21]), are inferring of relations, representing incomplete information, or even reasoning about evolutions of situations. However, we again stick to the claim that such an ontology-driven approach does not inhibit traditional approaches. For example, one could use fuzzy sets in order to cope with uncertain information from the lower levels

[4] Semantic Web Rule Language, http://www.w3.org/Submission/SWRL/
[5] http://jena.sourceforge.net

of information fusion and equip individuals of the ontology with membership measures during situation assessment.

To sum up, if the above application scenarios of ontologies are implemented for a SAW system, the system is indeed ontology-driven. However, ontology-driven does not mean that traditional approaches to SAW cannot be integrated. Rather, the SAW system and its users would benefit from having an ontology serve as the conceptual model of SAW and the advantages stated above.

3.6. Current Approaches

There are a number of domain-independent ontologies for SAW which we have evaluated in previous work (cf. [14]). The results of our evaluation indicated a feature which almost all ontologies neglected: The discussion about universally applicable relations, especially regarding the aspects of space and time. Moreover, to the best of our knowledge, we found just one approach that addresses all of the application scenarios stated above. SAWA, the Situation Awareness Assistant by Matheus et. al. [13], originates from the military domain and is a set of tools developed by a commercial company[6]. The basis of SAWA is a domain-independent ontology for situation awareness. In addition to OWL, SAWA employs SWRL for deriving relations among objects using rules. In SAWA, each situation type has a goal, the so-called 'standing relation', for constraining the number of relations which have to be determined. Although the standing relations are supposed to instantiate a situation, they can not be used for defining situation types as envisioned in subsection 3.4. Moreover, SAWA also misses universally applicable, spatio-temporal relations.

3.7. BeAware!

The shortcomings stated above have been the motivation for starting our research project BeAware! which features an ontology-driven framework for SAW system. BeAware! is based on the Jena Semantic Web framework[7] and combines OWL-DL reasoning capabilities and logic programming using Jena's generic rule reasoner. Whereas the basic concepts of BeAware!'s core ontology are similar to SAWA's ontology, we have extended it by primitive spatio-temporal relations from the fields of qualitative spatio-temporal reasoning (cf. [19], [20]). Thereby, we are in a position to define situation types and assess situations according to the application scenarios motivated above [21]. To show its real-world applicability, we implemented a proof-of-concept implementation for the RTM domain which we are currently deploying in the context of the Austrian highways agency's traffic management and information system[8]. In the following, we briefly describe the current experiences from implementing the above application scenarios.

1. *Integrate*—we developed the mappings from an information source to the SAW core ontology using Jena forward chaining rules, e.g. `(?a rdf:type rtm:Accident) -> (?a rdf:type saw:Object)` which means that if there is an instance of `Accident` from the RTM domain, the individual is also an in-

[6]Versatile Information Systems, Inc., http://www.vistology.com
[7]http://jena.sourceforge.net/
[8]ASFINAG, http://www.asfinag.at

stance of `Object` from the SAW core ontology. We followed such an instance-based approach rather than an implementation using OWL derived classes, because we had to transfer individuals from a source ontology to a target ontology for performance reasons. That is, the left-hand side of the above implication is evaluated in the source ontology, whereas the triple on the right-hand side is added to the target ontology. Although the above mapping is straight-forward, we met the issue of mapping heterogeneous elements between ontologies as outlined in [22]. Since there is no agreed approach how to construct such mappings, we developed a proprietary approach based on Jena rules which construct individuals on the right-hand side of the implication (e.g. in case we meet a property `rtm:isLocatedAt` in the source ontology, we construct an instance of `saw:Location` in the target ontology).

2. *Communicate*—although we have not yet integrated BeAware! into an existing user interface of a traffic operator, first presentations of a prototype have shown that the approach ontology browsers usually follow is not suitable for an integrated user interface; in fact, users should not be aware of the ontology in the background. The approach we currently follow is to provide understandable RDF comments which describe individuals and their classes and present them to the user. Moreover, just the relevant properties of an individual are presented leading to different views of the ontology depending on the kind of user (e.g. traffic operator, traffic engineer). The main advantage of using the ontology on the user interface is the possibility to drill down, i.e. the user interface may provide explanations for every assessed situation with minor implementation overhead.
3. *Exchange*—this application scenario is the last one scheduled and, thus, not yet implemented.
4. *Define*—discussions with traffic engineers have shown that they may cope with the rule- and ontology-based approach for defining situation types. Especially the different types of spatio-temporal primitive relations in the ontology, which make up the fundament for defing situation types, have been well received. An interesting figure is that current experiments lead to about 20 different situation types which suffice to assess most critical situations on the road network. This surprisingly small number of situation types is due to the fact that the taxonomy of objects in the RTM domain ontology enables quite generic situation types which cover the most common critical situations. In order to pinpoint particularly critical situations like a wrong-way driver, rather concrete situation types are defined. This example demonstrates the overall process of developing situation type definitions. One should start we generic and obvious situation types. During the operation of the system, these situation types should be refined and extended based on the experiences of the traffic operators.
5. *Assess*—Although we can currently not provide concrete measurements, the performance of handling the about 1000 traffic objects on Austrian's highways, which are constantly present in the domain ontology, are promising. BeAware! operates in an asynchronous way, i.e. updates to objects are pushed through a processing chain. Thus, our main focus is to leverage the performance of revisiting the objects affected by such an update. By exploiting the a priori knowledge about primitive relations, we are in a position keep the processing time of up-

dates at a constant rate. An exception is the initial situation assessment, which is actually not critical, but still needs some optimization.

4. Conclusions

In the scope of this paper, we have focused on application scenarios of ontology-driven SAW systems. The identified scenarios are the integration of heterogeneous information sources, the communication with the human operator, the exchange of knowledge about situations, the definition of situation types, and the actual situation assessment. We argue that the implementation of an ontology-driven SAW system based on a domain-independent SAW ontology resolves a shortcoming of traditional approaches to SAW, namely the missing formal conceptual model for SAW. In addition, such an ontology-driven SAW system entails all the advantages identified in the corresponding application scenarios without replacing traditional probabilistic approaches to SAW.

One problem regarding the implementation of this vision is the missing standardized rule language on top of OWL, because especially the definition of situation types and the exchange of knowledge about situations depend on an interoperable and universal standard. We hope that the results of the W3C RIF Working Group[9] will close this issue. Although our own framework BeAware! is already finished, the factual incorporation of traditional approaches to SAW is also an open issue. Potentially starting with fuzzifying relations, we are optimistic to implement the sketches outlined in this paper.

Regarding the commercial exploitation of BeAware!, we are currently in talks with a leading German manufacturer of RTM systems aiming at the integration of the framework into their product which will hopefully be another example for a successful industrial application of formal ontologies.

References

[1] James Llinas, Christopher Bowman, Galina Rogova, and Alan Steinberg. Revisiting the JDL data fusion model II. In *Proc. of the 7th Intern. Conference on Information Fusion, Stockholm, Sweden*, pages 1218–1230, June 2004.

[2] Heribert Kirschfink, Josefa Hernandez, and Marco Boero. Intelligent traffic management models. In *ESIT 2000 - European Symposium on Intelligent Techniques, Aachen, Germany*, AC-01, pages 36 – 45, Aachen, Germany, September 2000.

[3] M. R. Endsley. Design and evaluation for situation awareness enhancement. In *Proceedings of the Human Factors Society 32nd Annual Meeting*, pages 97–101, Santa Monica, CA, USA, 1988. Human Factors Society.

[4] Joachim Biermann. Challenges in High Level Information Fusion. In *Proceedings of the Seventh International Conference on Information Fusion, Stockholm, Sweden*, pages 526–527, June 2004.

[5] S. Das, R. Grey, and P. Gonsalves. Situation assessment via bayesian belief networks. In *Proceedings of the International Conference on Information Fusion*, pages 664–671, Annapolis, MD, 2002.

[6] Glenn Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, NJ, USA, 1977.

[7] M. M. Kokar and Z. Korona. A formal approach to the design of feature-based multi-sensor recognition systems. *International Journal of Information Fusion*, 2(2):77–89, 2001.

[9]Rule Interchange Format, http://www.w3.org/2005/rules/wiki/RIF_Working_Group

[8] Vladimir Gorodetsky, Oleg Karasaev, and Vladimir Samoilov. On-Line Update of Situation Assessment: A Generic Approach. *International Journal of Knowledge-Based and Intelligent Engineering Systems*, 9(4):361–365, November 2005.

[9] Mieczyslaw M. Kokar, Christopher J. Matheus, Kenneth Baclawski, Jerzy A. Letkowski, Michael Hinman, and John Salerno. Use Cases for Ontologies in Information Fusion. In *Proceedings of the Seventh International Conference on Information Fusion, Stockholm, Sweden*, June 2004.

[10] Thomas Strang and Claudia Linnhoff-Popien. A context modeling survey. In *Proceedings of the First International Workshop on Advanced Context Modelling, Reasoning and Management, Nottingham, England*, September 2004.

[11] M.A. Musen. *Knowledge Engineering and Agent Technology*, chapter Ontology-Oriented Design and Programming, pages 3–16. Frontiers in Artificial Intelligence and Applications. IOS Press, 2004.

[12] Ubbo Visser. *Intelligent Information Integration for the Semantic Web*. Springer-Verlag, Berlin, Germany, 2004.

[13] C. Matheus, M. Kokar, K. Baclawski, J. Letkowski, C. Call, M. Hinman, J. Salerno, and D. Boulware. SAWA: An assistant for higher-level fusion and situation awareness. In *Proc. of SPIE Conference on Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications, Orlando, USA*, pages 75–85, March 2005.

[14] Norbert Baumgartner and Werner Retschitzegger. A survey of upper ontologies for situation awareness. In *Proc. of the 4th IASTED International Conference on Knowledge Sharing and Collaborative Engineering, St. Thomas, U.S. VI*, pages 1–9, Nov. 2006.

[15] Amit Sheth, Kunal Verma, and Karthik Gomadam. Semantics to energize the full services spectrum. *Commun. ACM*, 49(7):55–61, 2006.

[16] Jon Barwise and John Perry. *Situations and Attitudes*. MIT Press, USA, 1983.

[17] Christopher J. Matheus, Mieczyslaw M. Kokar, Kenneth Baclawski, and Jerzy Letkowski. An application of semantic web technologies to situation awareness. In *Proceedings of the International Semantic Web Conference*, pages 944–958, 2005.

[18] Norbert Baumgartner, Werner Retschitzegger, and Wieland Schwinger. A software architecture for ontology-driven situation awareness. In *To appear in Proceedings of the 23rd Annual ACM Symposium on Applied Computing, Track Semantic Web and Applications, Fortaleza, Ceará, Brazil*, 2008.

[19] Norbert Baumgartner and Werner Retschitzegger. Towards a situation awareness framework based on primitive relations. In *Proceedings of the IEEE Conference on Information, Decision, and Control (IDC), Adelaide, Australia*, pages 291–295, February 2007.

[20] Norbert Baumgartner, Werner Retschitzegger, and Wieland Schwinger. Lost in time, space, and meaning—an ontology-based approach to road traffic situation awareness. In *Proc. of the 3rd Workshop on Context Awareness for Proactive Systems (CAPS), Guildford, UK*, June 2007.

[21] Norbert Baumgartner, Werner Retschitzegger, Wieland Schwinger, Gabriele Kotsis, and Christoph Schwietering. Of situations and their neighbors—evolution and similarity in ontology-based approaches to situation awareness. In *Proc. of the 6th International and Interdisciplinary Conference on Modeling and Using Context, Roskilde, Denmark*. Springer LNAI, Aug. 2007.

[22] C. Ghidini, L. Serafini, and S. Tessaris. On relating heterogeneous elements from different ontologies. In *Proc. of the 6th International and Interdisciplinary Conference on Modeling and Using Context, Roskilde, Denmark*. Springer LNAI, Aug. 2007.

Formal Ontologies Meet Industry
S. Borgo and L. Lesmo (Eds.)
IOS Press, 2008

Ontological Domain Coding for Cultural Heritage Mediation

Rossana DAMIANO [a], Vincenzo LOMBARDO [b], Fabrizio NUNNARI [c] and Antonio PIZZO [d]
[a] *Dipartimento di Informatica and CIRMA, Università di Torino, Italy; E-mail: rossana@di.unito.it*
[b] *Dipartimento di Informatica and CIRMA, Università di Torino; E-mail: vincenzo@di.unito.it*
[c] *VRMMP, Torino; E-mail: nunnari@vrmmp.it*
[d] *Dipartimento DAMS and CIRMA, Università di Torino; E-mail: antonio.pizzo@unito.it*

Abstract. An ontology-based representation of information about a domain is flexible enough to support different strategies in presenting the information. In this paper, we present two applications for creating interactive presentations in a cultural heritage domain, that share the same database of informative units and the same representation of the domain, encoded in a light-weight ontology.

An application for drama-based guided tours assembles the informative units in a location-aware fashion, by exploiting the structure of the ontology to enforce the notion of discourse focusing in the generated presentation. A browsing-based application for accessing the informative units supports semantic search, consulting the ontology to suggest modifications of the user's search to circumscribe or enlarge the result sets.

Introduction

Today, the ample support to interactivity provided by the multiplicity of available network infrastructures and user devices (desktop computers, mobile devices, phones) has propelled the development of personalized and adaptive applications for the access to information. In particular, in the domain of cultural heritage, the annotation of archives with semantic tags is the key to personalized access to information, as exemplified by the applications described by [1] and [11].

In this paper, we present two applications for creating interactive presentations in a cultural heritage domain. These applications are based on the same ontological representation of the domain, but realize two different strategies for mediating between the user and the domain. The informative items which constitute the knowledge base of the applications are encoded as self-contained audiovisual clips in which an artificial character, the anthropomorphized spider Carletto, presents specific information about a historical site (see Figure 1). The domain of the presentation consists of a historical location situated in Turin, Palazzo Chiablese, a 16th Century baroque palace that hosts the former royal apartments of the Savoy family. The royal apartments include five rooms:

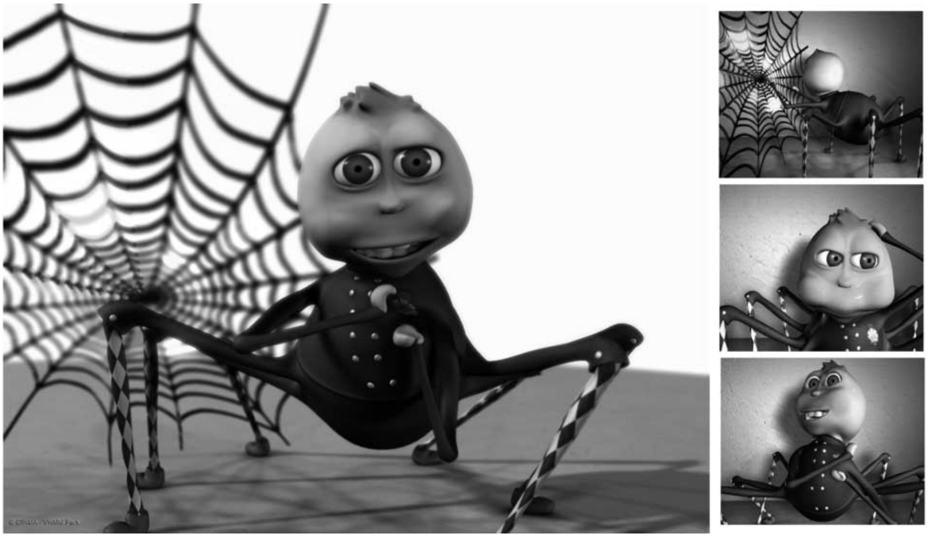

Figure 1. “Carletto the spider".

the Guardroom, the Room of the Valets, the Reception Room, the Dining Room, and the Room of the Tapestries. A domain expert (an historian of the art) has provided the information to be conveyed to the users. The input from the domain expert has been encoded in a set of micro-scripts by a drama expert, from which the audiovisual clips containing the presentation delivered by the artificial character have been produced (*presentation units*), with the help of a multimedia production team. The description of the domain has been encoded in an ontology with the assistance of an ontology engineer, providing the basis for the semantic tagging of the units. The production process is described in detail in [9].

The first application, ‘Carletto the virtual guide’, was designed to create guided tours in the palace for a public opening in April 2006. The presentation, delivered on a mobile device, is generated by selecting and sequencing the clips in a way that accounts for the position of the user in the historical site. The semantic coherence of the presentation is guaranteed by the fact that the system follows the relations encoded in the ontology to generate the presentation. The peculiarity of this application is that the presentation strategy is inspired to the principles of drama, with the goal of adding an emotional quality to the user experience (see [3] for the drama-based presentation paradigm).

In parallel with the guided visit, we designed a web-based application for direct access to the audiovisual clips, ‘Carletto the search engine’, by relying on the fact that the encoding of domain information in a ontological form provides a highly-structured semantic representation that supports personalized ways of accessing the same conceptual domain [4,2]. In this application, the user inserts a set of keywords, and a set of clips are retrieved based on the semantic match between the keywords and the concepts represented in the ontology. The structure of the ontology is also exploited to suggest modifications to the user’s search, by circumscribing or enlarging the result sets.

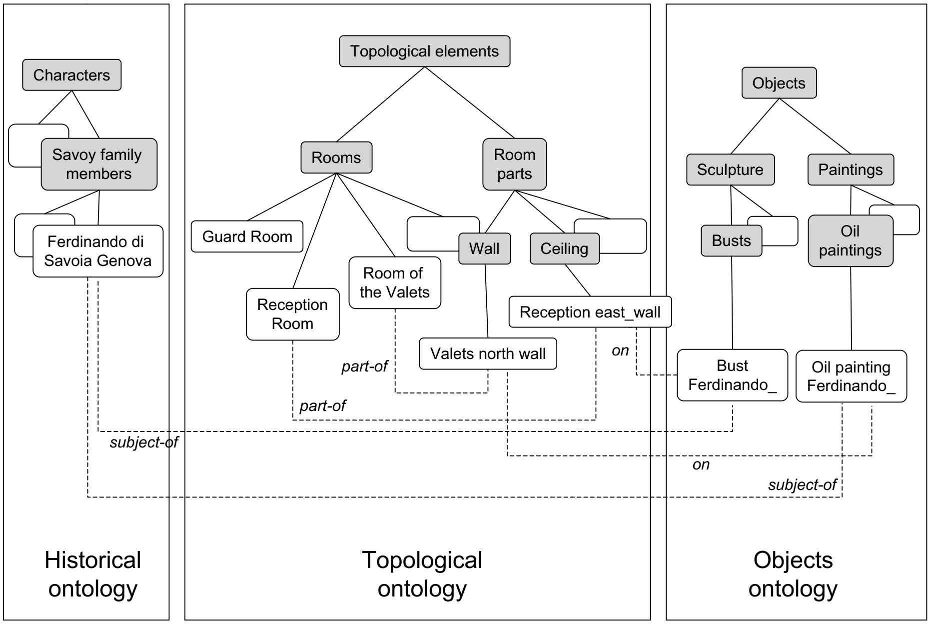

Figure 2. A fragment of the domain ontology concerning Palazzo Chiablese, including portions of three specific ontologies (historical, topological and object ontologies). Grey boxes represent classes, white boxes represent instances. Subclass and instance relations are represented by solid lines, non-taxonomic relations are represented by dashed lines.

1. Ontology-based representation of domain

The two applications described in this paper share the same repository of informative items (audiovisual presentation units). Units are self-contained, so a single unit can be delivered to the user without the risk of introducing unwanted references to other units.

Units have been tagged by their author with a description of their content through semantic metadata. Semantic metadata refer to a set of light-weight domain ontologies that encode the concepts needed to describe the domain (Palazzo Chiablese). Basically, the application for guided tours uses these metadata to sequence the units in a location-aware fashion and to manage the presentation focus in a coherent way given the current location. The search application, given the keywords provided by the user, exploits them to select a set of semantically relevant units based on a predefined mapping from keywords to concepts, and lets the user select among the proposed units.

Since the location of Palazzo Chiablese is a complex domain, that can be described according to several semantic dimensions (like history, art or topology), the representation of the domain is subdivided into five specialized ontologies (see Figure 2), and the topic of the presentation units is described as a tuple of references to these ontologies. This representation accounts for the fact that the same unit possibly concerns more than one topics, each described by a different ontology. A unit is constrained to refer to only one class or instance in each ontology, but may not refer at all to a certain ontology (provided that at least one ontology is referenced).

- The *topological ontology* describes the topology of location, centered on notions like rooms and room parts, according to the practice normally followed by human guides. Since both applications are intended to support the visit of the location in some way, topology provides a dimension of primary importance to organize the domain information.
- The *historical ontology* describes the historical facts related to the location. This ontology includes two main branches, describing respectively the historical characters who lived in the palace and those who worked in it (further subdivided into painters, architects and craftsmen).
- The *ontology of objects* systematizes the variety of pieces of furniture and other items located in the apartments, most of which are awkwardly termed and unknown to standard visitors.
- The *chronological ontology* is an ontology of time intervals that serves the purpose of providing a temporal framework for locating the historical events.
- The *symbolic ontology* describes the concepts (reigns, battles, marriages) that are celebrated by the art objects, located in the palace (paintings, statues, ect.).

In these ontologies, concepts are connected by subsumption relations in which each concept has only one ancestor, so all the ontologies are taxonomies. In order to simplify the description, the same concept cannot appear in more than one taxonomy.

In order to represent the non-taxonomic relations among concepts, orthogonal relations (different than subsumption) have been added to the domain description to connect concepts within the same ontology or across different ontologies. For example, the topological ontology contains a taxonomy of topological concepts (like rooms and room types) and the *part-of* relations according to which these entities are related, specifying, for instance, that a room contains a set of walls, a ceiling and a floor. As an example of relations spanning across different ontologies, consider the *subject of* a painting (object ontology), *located in* a room (topological ontology) and *painted by* an artist (historical ontology) to *celebrate* an event (symbolic ontology). The domain ontologies have been developed with the Protégé ontology editor; orthogonal relations have been implemented as Protégé slots [5].

2. Drama-based presentation

The core of the dramatization process consists in setting up, for the display to the user, some internal conflicts of the character, concerning its emotional values [8,10]. Conflict here is intended as a tool to help the user/visitor to build an emotional bond with the character. The visitor is part of the process of dramatic achievement, since during the presentation she/he provides feedbacks that are used to let the character's conflicts emerge. For example, if the visitor remains in the same room for a certain time, the virtual character switches from the role of a professional guide to the role of a 'storyteller', reflecting his personal involvement in the history of location.

The system architecture for the application "Carletto the virtual guide" is represented in Figure 3. It includes three cascading modules for managing, respectively, the interaction with the visitor (Interaction Manager), the presentation strategy (Presentation Manager), and the delivery of the audiovisual units (Delivery Manager).

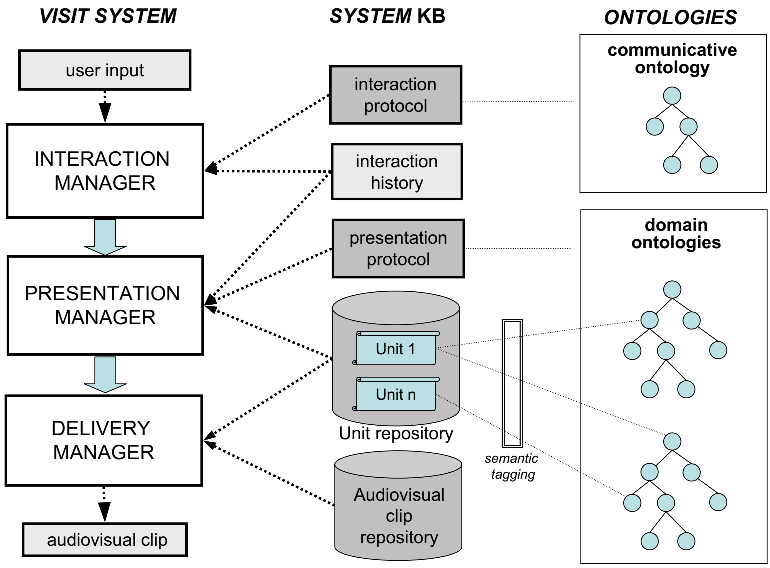

Figure 3. The architecture of the application “Carletto the virtual guide". Solid lines represent control flow, dashed lines represent data flow.

The strategy followed by the Interaction Manager to structure the interaction with the visitor and the adaptation to location is encoded in the *interaction protocol*, that specifies the communicative behavior of the character in terms of *communicative functions* [7] (encoded in a separate ontology). The activation of a function depends on the communicative context, given by the user input, the interaction history and the current location of the user.

In short, the interaction strategy of the application is the following. The visitor's actions of switching the PDA on or off have the highest priority and activate the functions that correspond to the opening and closing of the interaction. The termination of the visit (defined by the general visit requirements posed by the management of the historical site) has the immediately following priority, and results in the same effect of activating the closing function. If the location of the visitor is unknown, the character signals to the visitor that the location cannot be obtained, and invites the visitor to move in order to regain visibility (directive function). Finally, if the system detects that the visitor has remained in the current room for too long, it enters a waiting loop in which the character invites the visitor to a different room (directive function), and starts waiting for a reaction by alternating the character's idle state (phatic function) with new requests to move (e.g., Carletto polishes his medals and says impatiently “what about changing room?"). The notion of remaining in a room for too long is implemented as a threshold on the set of available presentation units for the room in the system knowledge base - a set that is decreased every time the visitor returns to the room.

If none of the previous cases occurs, the control passes to the Presentation Manager, that follows the location-adaptive *presentation protocol* represented in Figure 4. First of

```
1  PROCEDURE presentation_protocol (room, interaction history)
2     IF room has changed THEN
3        set current ontology to topological
4        set focus to room
5        select unit
6     ELSEIF focus set to room THEN
8        set focus to first room part
9        select unit
10    ELSEIF related unit on history exists THEN
11       set current ontology to historical
12       select unit
13       set current ontology to topological
14    ELSEIF delivery ratio of current room part is reached THEN
15       move focus to next room part
16    ELSE
17       select unit
18    ENDIF
19  ENDPROCEDURE
```

Figure 4. The presentation strategy of the application "Carletto the virtual guide".

all, the system checks if the visitor has moved to a new room (line 2). If so, it sets the ontology to the topological one (line 3) and sets the focus to the current room (the node of the ontology that corresponds to the room, line 4), so that a generic presentation unit about the room is selected (line 5, see below the selection criteria).

On the contrary, if the visitor is in the same room as the previous selection cycle, the system either starts or continues the description of the content of the room. By doing so, it is driven by a topological principle, i.e., by the order according to which items are positioned with respect to the four walls of the room and the ceiling (i.e., the room parts). If the focus was previously set at the room level (line 6), the system moves the focus to some room part (line 8).

At this point (line 11), the protocol realizes an alternation between the topological ontology and the historical ontology. This alternation aims at realizing the character's inner conflict between the roles of a guide and of a storyteller. When switching to the historical ontology, the selection of the subsequent presentation unit is driven by the attempt to minimize the transition from the topological ontology. So, the presentation units whose topic metadata include a reference to the historical ontology are searched for those that share the same topic as the current one for the topological (or historical) ontologies ("related unit on history", line 10). If no such unit is available, the systems searches for the units whose topic is a more general topic than the current one, and so on, until the top of the ontology is reached. If no such unit is available, the system skips the 'historical' detour.

After the ontology alternation has been dealt with, the algorithm checks whether there is any non-delivered information left to say about the current room part (line 14); otherwise it moves the focus to the next room part (the next sibling of the wall node on the ontology, line 15). Notice that the described strategy complies with the focussing rules stated by Grosz and Sidner [6] for task-oriented dialogue. According to Grosz and Sidner's rules, maintaining the focus on the current task has the priority over moving the

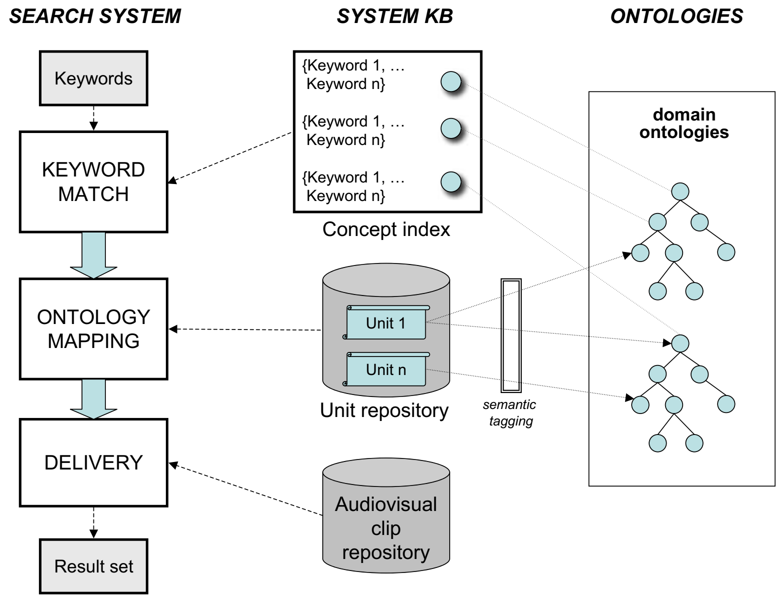

Figure 5. The architecture of the application “Carletto the search engine"; the architecture shares most of the system kb and domain ontologies with the application “Carletto the virtual guide" (Figure 3).

focus to a subtask of the current task, that has the priority over moving the focus from the current subtask to a different subtask of the same task. [1]

The visit server and client are implemented in Java (http://java.sun.com), while the clip repository is implemented as a mySql data base (http://www.mysql.com). The client software is executed by a virtual machine on a PDA (ASUS A636 PocketPC series).

3. Keyword-based search

Beside the system for creating character-based, dramatized guided tours, we designed an application in which the same repository of audiovisual units can be accessed through a simple semantic search engine. In this application, the user enters a set of keywords to perform a search on the repository, possibly retrieving a non empty set of units. The user can ask the system for suggestions to improve the search through narrowing or widening. The architecture of this application, “Carletto the search engine", is sketched in Figure 5.

In the search application, the repository does not include the units devoted to the management of the interaction with the user (for example, the units in which the virtual character greets the visitor – without providing any domain information), since the interaction with the user is limited to the insertion of a set of keywords in the search interface.

[1]The actual implementation supports the focussing procedure on a taxonomy of any depth. However, here we report the algorithm only for the levels of the room and the room parts, since only these two were developed in the final version of the system.

```
FUNCTION search (keywords list)
   FOREACH keyword IN keyword list
      IF keyword matches ontology node THEN
         add node to matching node set
         FOREACH node IN matching node set
                 IF related units exist THEN
                    add units to result set
                 ENDIF
         ENDFOREACH
      ENDIF
   ENFOREACH
   return result set
 ENDFUNCTION
```

Figure 6. The search strategy of the application “Carletto the search engine".

Given keywords entered by the user, the system consults an index in which the nodes of the domain ontologies are mapped to a set of keywords, selected by an interaction designer. If one or more matches are found, the system queries the repository of audio-visual units for the units that are related with the matching nodes through their semantic metadata.[2] The process is illustrated in Figure 6.

The selected units are proposed to the user as a list of links to the clips, that the user can access in different modalities (text or multimedia) and formats. A priority is given to the matches with the topological ontology, given the relevance of this dimension. When the user selects a unit from the list, the system also displays links to the other concepts to which that unit is related, extracted from the metadata of the unit itself. If the user clicks on one of these links, the system searches the repository with the selected concept as new matching node; through this modality, the user can browse the repository as a hypertext.

If the user wishes to broaden or restrict the search, the system can make suggestions following the hierarchical relations in the ontologies (Figure 7). To broaden the search, the system repeats the query on the unit repository by replacing the matching nodes in the node list with their direct ancestors (if the nodes are instances, the instances are replaced by the class to which they belong). On the contrary, to narrow the search, the system replaces each node with the set of its children. If explicitly requested by the user, the system can also query the unit repository for all the units that are related with the node list via subsumption, bypassing in one step the iteration of the narrowing process until the leaves of the taxonomies.

This application is currently being prototyped in Java by exploiting on the tools made available by Protégé for the access and manipulation of the ontology.

4. Examples

In order to exemplify the role of the ontologies in the two applications, we resort to an example based on the portion of ontology shown in Figure 2. The figure shows some fragments of the historical, object and topological ontologies; as it can be noticed by observing the figure, the node representing the historical character of the Duke Ferdi-

[2] Units that are not *directly* connected with the matching nodes are not considered, following the assumption that the level of abstraction selected by the user must be preserved.

```
FUNCTION broaden-search (node set)
   FOREACH node IN node set
      replace node with ancestor
      FOREACH node IN node set
              IF related units exist THEN
                 add units to result set
              ENDIF
      ENDFOREACH
  ENDFOREACH
  return result set
ENDFUNCTION
```

```
FUNCTION narrow-search (node set)
   FOREACH node IN node set
      replace node with descendants
      FOREACH node IN node set
              IF related units exist THEN
                 add units to result set
              ENDIF
      ENDFOREACH
  ENDFOREACH
  return result set
ENDFUNCTION
```

Figure 7. The algorithms for broadening and narrowing the scope of the search in the semantic search application.

nando di Savoia Genova (one of the owners of the Palace, instance of the class "Savoy family members") is connected to some instances of the object ontology by a relation of the type “subject_of". In fact, two of the busts contained, respectively, in the apartments (Reception room) and in the entrance hall of the Palace, represent the character of the Duke Ferdinando. The Duke Ferdinando is also depicted in an oil painting contained in the Room of the Valets.

In “Carletto the virtual guide", Carletto is likely to talk about the Duke Ferdinando at different points in the presentation (due to location adaptation and to random aspects in the presentation, presentations vary from a visitor to another, and from a session to another with the same visitor). When the system follows a topological order of presentation, the Duke Ferdinando may be mentioned when Carletto is talking about the locations in which the painting and the busts are located. The Duke Ferdinando may subsequently become the subject of a historical digression centered on the Savoy family.

So, the presentation flow may be constituted by the following sequence of presentation units (taken from one of system logs):

4 $\mathtt{PU\_33} < \mathtt{Topology} : \mathtt{Room\_of\_the\_Valets}, \mathtt{History} : \mathtt{Carlo\_Felice} >$

...

7 $\mathtt{PU\_35} < \mathtt{Topology} : \mathtt{Room\_of\_the\_Valets\_north\_wall},$
$\mathtt{Objects} : \mathtt{oil\_painting\_Ferdinando}, \mathtt{History} : \mathtt{Ferdinando\_di\_Savoia} >$

8 $\mathtt{PU\_107} < \mathtt{Topology} : \mathtt{Palace}, \mathtt{History} : \mathtt{Ferdinando\_di\_Savoia},$
$\mathtt{Chronology} : \mathtt{1850}, \mathtt{Symbols} : \mathtt{Marriage} >$

With PU_33, the system starts introducing the room of the Valets by giving general information about its function and the character who gave it its current shape (Carlo_Felice), then illustrates the room walls one by one (for brevity, we omitted the two units about the south and east walls, represented by the dots). After presenting the oil painting on the north wall of the room (PU_35), the system switches to the historical digression. So, Carletto starts talking about the Duke Ferdinando (PU_107), with a unit that talks about his marriage with an Austrian princess, that led some main renovations in the Palace. Remember the system tries to minimize the transition between the ontologies by looking for a unit that shares the most of the values in the semantic metadata with the last selected one: in this case, the closest match for the unit PU_35 is the unit PU_107, that has the same value for the historical ontology (see the item `History` : `Ferdinando_di_Savoia`).

Alternatively, the system may introduce the Duke Ferdinando when presenting the Reception room, which contains a bust that represents him. Or, he may be mentioned in a historical digression after Carletto talked about the Duke's marriage, since this event is explicitly mentioned in one of the units concerning the Reception Room.

In the application, "Carletto the search engine", if the user inserts the keyword "Ferdinando di Savoia Genova", the system finds a set of presentation units that refer to the Duke Ferdinando in their semantic metadata. As shown above, some talk about him as a historical character (PU_106, PU_107), who played a role in the history of the Palace, some mention him as the subject of artworks (PU_58, PU_35). In order to broaden the search, the system would propose the user to use the keywords "Savoy" or "Savoy family", that are associated with the class representing the Savoy family, of which the Duke Ferdinando is a member; in this case, a larger set of results would be proposed (25 units). The search can be further broadened by pointing to the node that refers to the historical characters in general (yielding 34 units).

If the user cliks on the link associated with the word 'bust' in the text of the PU_58, the system shows all the units concerning the busts contained in the palace (PU_58, PU_31, PU_27, PU_48). Or, if the user cliks on the link associated with the Reception Room, the system retrieves all the units in which Carletto talks about that room (7 units).

5. Discussion

The two applications described in this paper rely on the semantic annotation of a repository of informative items (presentation units), which take the form of scripted audiovisual units. In both applications, items are retrieved and delivered to the user based on their semantic annotation in a context-dependent way, although the role of the context varies from the location-awareness of the virtual guide (Section 2) to the keyword-driven search in the search application (Section 3). A common feature underlying the two approaches is that the processes of writing the units that compose the presentation and the process of semantically annotating them are integrated in one process of "writing & tagging", accomplished by the author who writes the units. In both cases, the semantic metadata that the system relies on to retrieve the units are predefined with respect to the process of writing the presentation contents and constitute a framework by which the author is constrained and driven in the writing process, as described in [9]. This approach, although it poses important limitations to the author, is motivated mainly by practical considera-

tions: by giving a rich semantic characterization of the presentation units (the units are annotated according to a set of light-weight domain ontologies, as illustrated in Section 1), the system is not required to perform complex ontological reasoning, but navigates the domain ontologies by relying on the author's annotations.

A major drawback of this approach is that the effectiveness of the systems depends on the author's expertise and competence in tagging. For example, consider a presentation unit about a painting: the semantic metadata of that unit point to a certain instance of 'painting' in the object ontology, but if the the unit also mentions the author of the painting (modeled by the *painted by* relation between the painting instance in the object ontology and an author instance in the historical ontology), the unit metadata concerning the historical ontology should point to that specific painter instance. This style of representation makes some basic forms of ontological reasoning unnecessary for the system that accesses the presentation units; going back to the example before, it makes it superfluous for the system to query the ontology in search for a filler of the *painted by* relation between paintings and painters for a specific painting, as this information is likely to emerge from the manual encoding of the semantic metadata. The consistency of the manually-coded semantic metadata with the relations between the ontological representation of the domain is not guaranteed, since it is not governed by formal rules; rather, it is left to the initiative and the skills of the author who writes and tags the units: if the unit in the example marginally mentions a different painter than the painting author, it is not advisable – though not forbidden – to encode the reference to this other painter in the tuple.

The two systems presented here permit the data and their annotation to be modified to a limited degree. In particular, in the virtual guide application, the presentation script that drives the location-aware delivery of the information relies on the topological ontology and on the historical ontology to create a dramatic conflict in the virtual guide. In addition, the presentation script assumes some predetermined levels of detail in the description of the location (palace, room, room parts and objects), matching the structure of the topological ontology, and dictated by the design goal of mimicking the 'situatedness' that characterizes the visit with real guides. So, while adding new rooms or objects to the visit would not affect the system, designed and implemented to be scalable, transporting the system to a domain in which the topology is structured in a different way would require the presentation script to be modified. On the contrary, the search application is not influenced by the content of the domain ontologies, since it mainly relies on the navigation of taxonomic ontologies.

Concerning the evaluation of the domain coding adopted for the two applications, an evaluation study, conducted during the opening to the public of the virtual guide system, has revealed that the visitors appreciated the applications and were satisfied with the quantity and quality of the information provided (respectively, with a score of 3.80 and 3.72 on a scale from 0 to 5), and that they were able to find the items addressed by the virtual guide (83% of the visitors). Although the evaluation did not specifically address the appropriateness of the ontologies employed by the application, the overall appreciation for the system seems to confirm that they were suitable to represent the domain for the application purposes, and that the focusing mechanism that they supported was adequate to establish an effective reference system for the user.

6. Conclusions

In this paper, we have presented two applications for creating presentations that mediate between users and domain information in the field of cultural heritage. Although inspired by different design goals – the dramatized presentation system aims at improving the user's reception of the information by generating an engaging guided tour, while the web-based application is a simple search interface – the two applications share the same ontological representation of the domain. This representation, necessary for accounting for the complexity of the presentation domain (a historical location), lends itself to developing different paradigms to access the information, allowing at the same time the process of reusing data among different applications.

References

[1] L. Aroyo, R. Brussee, P. Gorgels, L. Rutledge, N. Stash, and Y. Wang. Personalized museum experience: The rijksmuseum use case. In *Museums and the Web*, San Francisco, CA, 11-14 April 2007.

[2] Melike Şah, Wendy Hall, Nicholas M. Gibbins, and David C. De Roure. Semport: a personalized semantic portal. In *HYPERTEXT '07*, pages 31–32, New York, NY, USA, 2007. ACM.

[3] R. Damiano, V. Lombardo, F. Nunnari, and A. Pizzo. Dramatization meets information presentation. In *Proceedings of ECAI 2006*, Riva del Garda, Italy, 2006.

[4] Susan Gauch, Jason Chaffee, and Alexander Pretschner. Ontology-based personalized search and browsing. *Web Intelligence and Agent Systems*, 1(3-4):219–234, 2003.

[5] J.H. Gennari, M.A. Musen, R.W. Fergerson, W.E. Grosso, M. Crubézy, H. Eriksson, N.F. Noy, and S.W. Tu. The evolution of Protégé: an environment for knowledge-based systems development. *International Journal of Human-Computer Studies*, 58(1):89–123, 2003.

[6] B. J. Grosz and C. L. Sidner. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12:175–204, 1986.

[7] N. Maudet and B. Chaib-draa. Trends in agent communication language. *Computational Intelligence*, 18 (2):89–101., 2002.

[8] R. McKee. *Story*. Harper Collins, New York, 1997.

[9] F. Nunnari, V. Lombardo, R. Damiano, A. Pizzo, and C. Gena. The canonical processes of a dramtized approach to information presenation. *Multimedia Systems. Special Issue on the Canonical Processes of Multimedia Production*, forth.

[10] L. Seger. *Creating Unforgettable Characters*. Henry Holt and Company, New York, 1990.

[11] J. van Ossenbruggen, A. Amin, L. Hardman, M. Hildebrand, M. van Assem, B. Omelayenko, G. Schreiber, A. Tordai, V. de Boer, B. Wielinga, J. Wielemaker, M. de Niet, J. Taekema, M. van Orsouw, and A. Teesing. Searching and annotating virtual heritage collections with semantic-web techniques. In J. Trant and D. Bearman, editors, *Museums on the Web*, San Francisco, CA, 11-14 April 2007.

Formal Ontologies Meet Industry
S. Borgo and L. Lesmo (Eds.)
IOS Press, 2008

An ontology for Environmental and Health and Safety risks' evaluation for construction

Alba FUERTES [a,1], Miquel CASALS [a], Marta GANGOLELLS [a], Nuria FORCADA [a] and Xavier ROCA [a]

[a] *Department of Construction Engineering, Technical University of Catalonia, Spain*

Abstract. The present research project establishes the basis of a decision making tool, based on an ontology, to integrally analyze Environmental and Health and Safety risks along the planning and execution phases of a construction project and to define technical solutions and preventive measures. The ontology will allow the classification of all the terms (aspects, impacts, risks, and procedures) related to the Health and Safety Evaluation and to the Environmental Evaluation as well as the relationships that exist among them. On the other hand, each class will be enriched with different properties that will be used by the decision-making tool to identify the main significant Environmental and Health and Safety aspects in each Construction process, and moreover, to evaluate their impact in a specific construction project in order to provide procedures. Therefore, both designers and contractors will asses the environmental and the health and safety risks related to each designed solution or constructive process, facilitating them the choice of the designed solution with fewer impact/risk. As a consequence the construction sector will increase its competitivity by optimizing the resources dedicated to the different management systems of a construction company.

Keywords. risks' evaluation, health and safety, environment, ontology, construction

Introduction

Historically, the construction sector has always been seen as a traditional and conservative industry. Nevertheless, from the beginning of the 21st century, technological advances have changed the way of working, leading to revisions and modifications of construction processes and practices. Market changes, new technologies and increasing client expectations are stimulating radical revisions on how the construction industry works and how to give value to its processes.

The aim of the project called "Development and implementation of a Web based Project Management System for the integrated Safety and Environmental Management in the production processes of the construction sector SMEs" is to help the construction sector to increase its competitivity by optimizing the resources dedicated to the different management systems of a construction company. Therefore, these companies usually use quality management systems on site; they sometimes apply safety and environmental management systems but seldom, if ever, apply integrated management

[1] Alba Fuertes: Research Engineer. Technical University of Catalonia. Department of Construction Engineering, C/Colom 11, Edifici TR-5, 08222 Terrassa (Barcelona) Spain. Email: alba.fuertes@upc.edu

systems. It habitually leads to a waste of money and human resources, which is avoidable by an application of a correct working methodology.

For this reason, in this project, a working methodology and an implementation model for an integrated management (especially in terms of safety and environmental management) is being developed. This project is focused on those companies who normally pay less attention to processes management: SMEs.

Moreover, we are developing a web application taking advantage of the existing information systems (Internet, wireless network, mobile equipment, etc.). This tool will provide the information access during all the construction process in order to assess, in an integrative way, the environmental and the health and safety aspects related to a construction project. Thereby, we are promoting, on one hand, the introduction of information and communication technologies in the construction field and on the other, the integration of the existing information systems both in the construction site and in the technical department. The web application will be designed to use ontologies in a collaborative environment.

1. Background

Construction professionals need to know how to balance the contingencies of risk with their specific contractual, financial, operational and organizational requirements. In order to achieve this balance, proper risk identification and risk analysis is required. The risk management process entails identifying construction risks and exposures, and formulating an effective risk management strategy to mitigate the potential for loss. Risk is a natural part of any business enterprise; however, the construction industry faces more than its fair share of risk factors. The list of risks is virtually endless and so is the list of risk management strategies that construction owners often deploy - reactively for the most part - during their projects.

Currently, there are different management models that construction companies are adopting to increase their competitivity. From one side, ISO 14000 [1] and Eco-Management Audit Scheme Regulation (EMAS) [2] are a reference point for environmental management and the basis for some construction companies to minimise the environmental impacts related to the construction process. On the other side, the OHSAS 18001 regulation [3] is the international occupational health and safety management system specification.

Many research works focus on the evaluation of the level of use of environmental and/or health and safety management systems in the construction sector ([4], [5], [6], [7], [8], [9], [10], [11], [12]). They also focus on how the systems are integrated, but in general, they highlight that this integration is limited to documental integration, which only means policies, manuals, procedures and registers unification.

The project we are exposing in this paper deals with the integrated evaluation and analysis of Environmental and Safety & Health Risks with the aim to control environmental and safety risks and reduce their impact.

2. Aim and methodology

The aim of this paper is to present a project which is currently being developed to help the construction sector to increase competitively by optimizing management systems

resources (Safety & Health and Environmental management systems) of the construction companies.

The specific objectives of this project are:

- To develop an ontology that represents the Concept Model built from all the terms related to the Health and Safety Evaluation (aspects, impacts, risks, procedures) and to the Environmental Evaluation (aspects, impacts, risks, procedures) as well as the relationships that exist among them.
- To establish the basis of a decision making tool, based on the previous ontology, to integrally analyze Environmental and Health and Safety risks along the planning and execution phases of a construction project and to define technical solutions and preventive measures.

3. Development of the Ontology for Environmental and Health and Safety risks' evaluation

3.1. Ontology Domain

This research attempts to develop an ontology focused on all the terms (aspects, impacts, risks, procedures) related to the Health and Safety Management and to the Environmental Management as well as the relationships that exist among them, to integrally analyze both impacts along the planning and execution phases of the construction project.

3.2. Concept Model

A Concept Model is developed in order to understand the main structure of the domain and to facilitate the development of the ontology. Figure 1 shows the important terms and their relationships that should be included in the ontology.

This Concept Model is based on the *Construction Activities* that take place along the execution phase of a construction project. Each of these Construction Activities is simultaneously related to both *Environmental Aspects* (defined by the Eco-management and audit scheme regulation (EMAS)) and *Health and Safety Aspects* (defined by the National Institute of Safety and Hygiene at work (INSHT)).

As a result of these interactions the *Environmental Risks* and the *Health and Safety Risks* are obtained. Moreover each of these impacts and risks has some related properties. From their combination and analysis, the Environmental and Health and Safety valuation are reached. As a consequence and from the study of the valuation, *Environmental alternative technical solutions* and *Health and Safety alternative technical solutions* are given, in order to reduce the Environmental Risks and the Health and Safety Risks, respectively.

On the other hand, these alternative technical solutions are applied along the Construction Activities prompting a direct effect to both Environmental and Health and Safety Aspects, as a consequence of the existing relation between the Construction Activities and both aspects. For example, if after conducting the assessment of certain construction project, the generation of dust in activities with construction machinery and transport is found to be an extremely significant impact, therefore, an environmental procedure related to proper watering must be established. Obviously,

this environmental procedure will affect the assessment of the environmental impact "Water consumption during the construction process".

As a result, all the important terms included in the Environmental and Health and Safety Evaluation domain are defined and related allowing an integral evaluation.

The proposed Concept Model is shown in Figure 1.

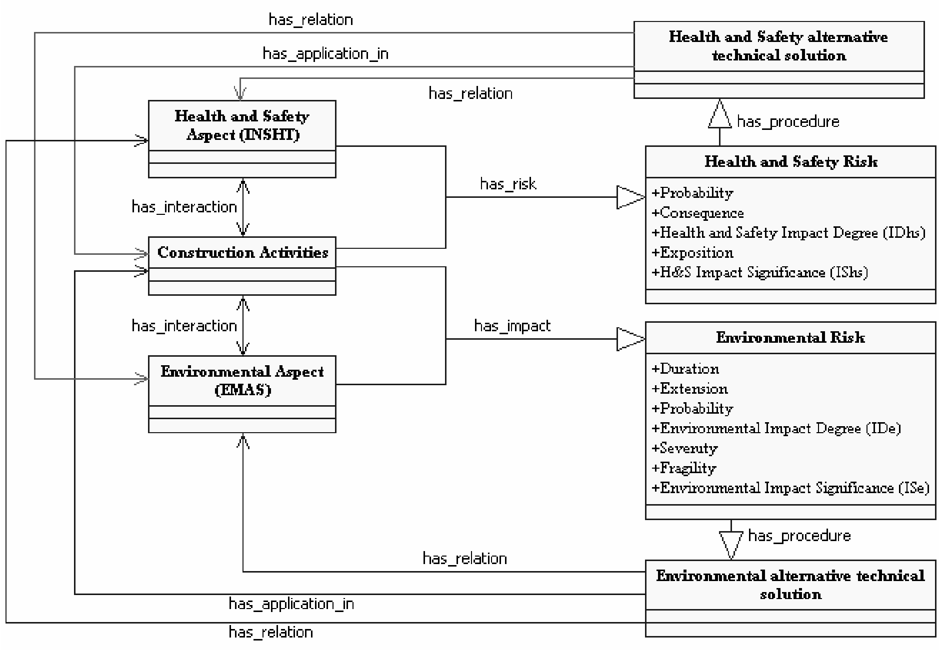

Figure 1. Concept Model

3.3. Implementation

The implementation of the ontology has been carried out in Protégé editor, for its opened code, free access and simplicity, working with OWL DL language.

Reusing ontologies is an important point to bear in mind before the development of a new ontology. In order to increase the interoperability and standardization most of the terminology used to describe the terms included in the ontology has been extracted from existing standards and regulations ([1], [2]).

From the concepts identified in the Concept Model, classes, subclasses and properties are defined. In this proposed ontology the classes and their subclasses are:

- *Construction Activities*, composed by all the activities that take place along a construction project.
- *Health and Safety Aspect*, composed by all the health and safety risks identified by the National Institute of Safety and Hygiene at work (INSHT).
- *Environmental Aspect,* composed by all the aspects defined by the Eco-management and audit scheme regulation (EMAS).

- *Health and Safety Risk*, composed by all the health and safety risks identified along a particular construction process.
- *Environmental Risk*, composed by all the environmental risks identified along a particular construction process.
- *Health and Safety alternative technical solution,* composed by all the procedures that should be used to solve the already identified Health and Safety Risks.
- *Environmental alternative technical solution*, composed by all the procedures that should be used to solve the already identified environmental risks.

To classify the documents in the created hierarchical structure properties of classes are defined. Three kinds of Protégé properties have been used:

- Object-properties are used to relate classes.
  - *has_interaction* relates the class Construction Process to the classes Health and Safety Aspect and Environmental Aspect.
  - *has_risk* relates the Health and Safety Aspect and the Construction Process classes with the Health and Safety Risk class.
  - *has_impact* relates the Environmental Aspect and the Construction Process classes to the Environmental Impact class.
  - *has_procedure* relates the class Health and Safety Risk with the class Health and Safety Procedure. In the same way, it also relates the class Environmental Impact with the class Environmental Procedure.
  - *has_relation* relates the class Health and Safety Procedure with both Health and Safety Aspect and Environmental Aspect classes. In the same way, it also relates the class Environmental Procedure with both Health and Safety Aspect and Environmental Aspect classes.
  - *has_application_in* relates the Health and Safety Procedure class and the Environmental Procedure class with the Construction Process class.
- Datatype-properties add information to the classes.
  When identifying specific environmental aspects related to the building construction process, three components of significance are considered:
  - *Extension,* can have any whole number value (int range).
  - *Probability,* can have any whole number value (int range).
  - *Duration,* can have any whole number value (int range) .
  - *Environmental Impact Degree (ID_E)* for a certain construction stage is obtained by combining all these three significance components. An environmental impact for a specific construction stage is considered to be significant if its ID_E is higher than a certain numerical value.

  Once significant environmental aspects for construction activities are obtained they have to be assessed depending on each specific building site.
  - *Severity*, relevance of an environmental aspect. Can have any whole number value (int range)
  - *Fragility,* can have any whole number value (int range).
  - *Environmental Impact Significance (IS_E)* for a certain construction project is obtained by multiplying both previous parameters. When IS_E is higher than a certain numerical value, it is possible to provide procedures to mitigate adverse impacts.

By the same way, when identifying specific health and safety aspects related to the building construction process:

- *Consequence*, can have any whole number value (int range).
- *Probability,* can have any whole number value (int range).
- *Health and Safety Impact Degree (ID_{HS})*, obtained by combining both previous values. A health and safety aspect for a specific construction stage is considered to be significant if its ID_{HS} is higher than a certain numerical value.

Once significant health and safety aspects for construction activities are obtained they have to be assessed depending on each specific building site.

- *Exposition,* component of significance ranging from 0 (non-existing risk) to 25 (extremely significant risk exposition).
- *Health and Safety Impact Significance (IS_{HS}),* assumed to be the equal to Exposition. When IS_{HS} is higher than a certain numerical value, it is possible to provide procedures to mitigate adverse impacts.

- Annotation-properties would be used to provide multi-lingual names for ontology elements.

4. Development of the Decision-Making Tool

The previously defined ontology serves as a basis for the development of a Decision-Making tool oriented to provide the information access during all the construction process in order to assess, in an integrative way, the environmental and the health and safety aspects related to a construction project.

This tool will facilitate decision-making and continuous improvement with an integrated conception of the whole constructive process. Currently, environmental and safety and health aspects are not taken into account until the construction stage. The proposed tool will evaluate these risks in different stages of the project, from the study stage to the planning and preparation stage. On one hand, designers will be able to evaluate the environmental risks and the health and safety risks related to each designed solution. On the other hand, contractors (who normally receive the building specifications, drawings, bill of quantities, health and safety plan, etc. of a project) will be able to asses the environmental impacts and the health and safety risks related to each constructive process.

The environmental risks and the health and safety risks will be also assessed during (inside and between) the different phases of the construction stage and at the end of the project taking into account the constructive process defined by the contractor. Interrelations between the application of technical solutions / preventive measures and resulting environmental risks and health and safety aspects will be considered.

As a result, different designed solutions and constructive processes will be evaluated and the ones with fewer impacts/risks will be chosen. As end users of this decision-making tool are basically designers, contractors, health and safety and environmental technicians, a collaborative environment (Computer-supported Cooperative Work - CSCW) is needed.

The developed model helps to evaluate environmental and health and safety risks (controlling the operations between and inside every phase), propose technical solutions and preventive measures and capture knowledge of the project and reuse it in other one.

Figure 2 shows the general layout of the decision making tool for Environmental and Health and Safety Risks management.

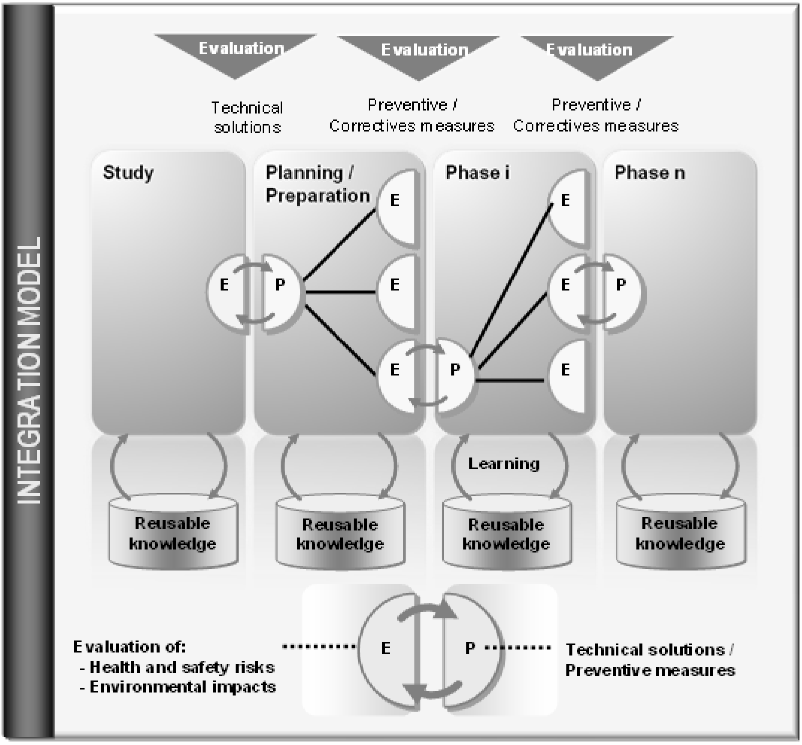

Figure 2. General layout of the Decision-Making tool

In each stage of the life cycle of a construction project, one or more environmental and health and safety evaluations should be carried out. This means, that the web application needs some screens to help the practitioners in entering the information related to the construction project (see prototype sketch in figure 3).

The Decision Making tool will provide an integrated Environmental and Health & Safety analysis to evaluate both risks in conjunction and will be able to extract technical solutions and preventive measures considering both risks (see figure 4).

Once the risks evaluation is carried out, the tool will propose technical solutions and modifications of the constructive process (see prototype sketch in figure 4). If it is not possible to modify the process, the tool will provide environmental and safety and health preventive measures. Then, when the construction stage has started and problems occurred, the tool will provide corrective measures.

The decision-making tool will also capture all the data from different construction projects so as to be able to reuse this information into the web application.

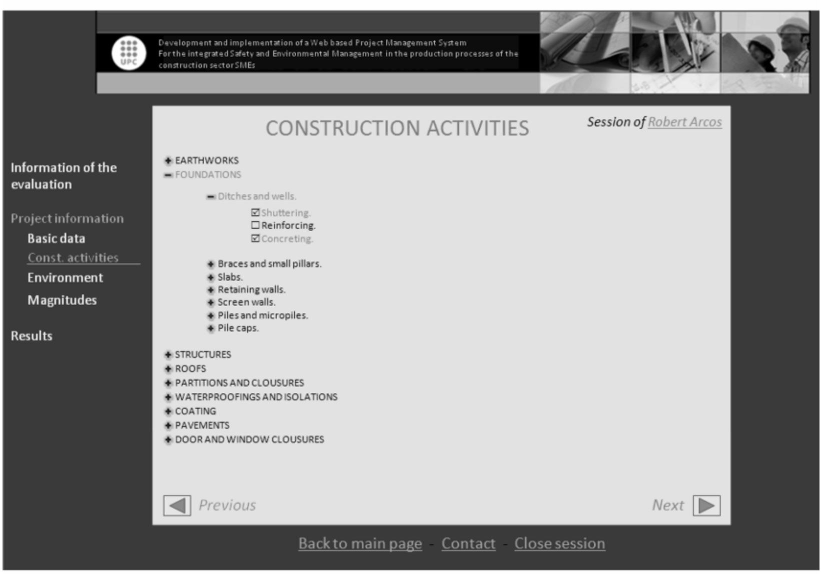

Figure 3. Screen of the prototype sketch showing the data entry

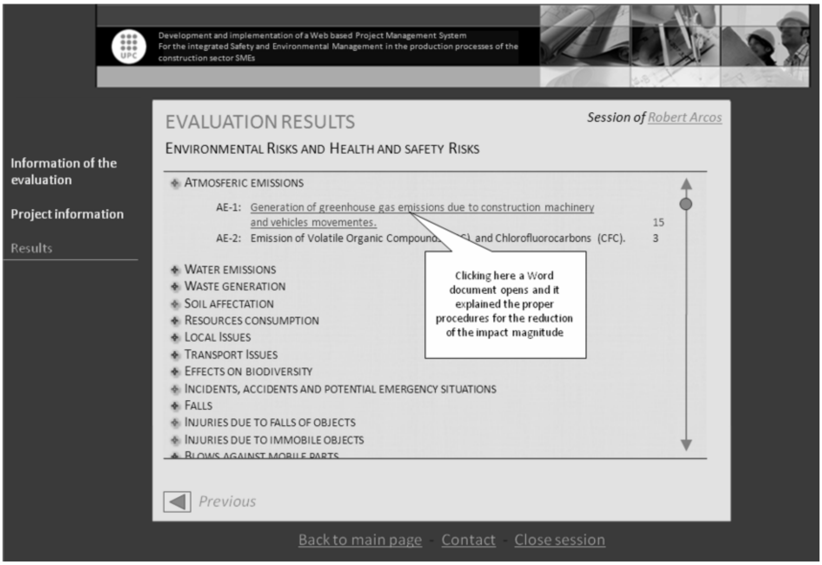

Figure 4. Screen of the prototype sketch showing the significance of environmental and health and safety risks of a certain construction project

5. Conclusions

The present research project establishes the basis of a decision making tool, based on an ontology, to integrally analyze Environmental and Health and Safety risks along the planning and execution phases of a construction project and to define technical solutions and preventive measures.

The ontology will allow the classification of all the terms (aspects, impacts, risks, and procedures) related to the Health and Safety Evaluation and to the Environmental Evaluation as well as the relationships that exist among them. On the other hand, each class will be enriched with different properties that will be used by the decision-making tool to identify the main significant Environmental and Health and Safety aspects in each Construction activity, and moreover, to evaluate their impact in a specific construction project in order to provide procedures.

Therefore, both designers and contractors will asses the environmental and the health and safety risks related to each designed solution or constructive process, facilitating them the choice of the designed solution with fewer impact/risk.

For the moment, the ontology and the tool are still being developed but it is expected that in the near future validation and verification by analyzing different case studies will be carried out.

6. References

[1] International Standardization Organization (2004) Environmental management systems - Requirements with guidance for use ISO 14001:2004

[2] EMAS Europa. Regulation (EC) No 761/2001 of the European Parliament and of the Council of 19 March 2001 allowing voluntary participation by organisations in a Community eco-management and audit scheme (EMAS).

[3] OHSAS (1999) Occupational Health and Safety Management Systems Specifications OHSAS 18001:1999

[4] Ofori G., Gang G., Briffett C. (2002). "Implementing environmental management systems in construction: lessons from quality systems". Building and Environment. Volume 37, Issue 12, pp. 1397-1407.

[5] Lam K.C., Ng S.T. (2006)"A cooperative Internet-facilitated quality management environment for construction". Automation in Construction. Volume 15, Issue 1.

[6] Shohet I.M., Frydman S. (2003). "Communication Patterns in Construction at Construction Manager Level". Journal of Construction Engineering and Management. Volume 129, Issue 5, pp. 570-577.

[7] Cheung S.O., Suen H.C.H., Cheung K.K.W. (2004). "CSHM: Web-based safety and health monitoring system for construction management". Journal of Safety Research. Volume 35, Issue 2, pp. 159-170.

[8] Pheng L.S., Tan J.H.K. (2005). "Integrating ISO 9001 Quality Management System and ISO 14001 Environmental Management System for Contractors". Journal of Construction Engineering and Management. Volume 131, Issue 11, pp. 1241-1244.

[9] Koehn E., Datta N.K. (2003). "Quality, Environmental, and Health and Safety Management Systems for Construction Engineering". Journal of Construction Engineering and Management. Volume 129, Issue 5, pp. 562-569.

[10] Guerra J.F., Moy D. (2003). "The new performance evaluation methodology and its integration with management systems". TQM Magazine. Volume 15, Issue 1, pp. 25-29.

[11] Mackau D. (2003). "SME integrated management system: a proposed experiences model". TQM Magazine. Volume 15, Issue 1, pp. 43-51.

[12] Zeng S.X., Tian P., Shi J.J. (2005). "Implementing integration of ISO 9001 and ISO 14001 for construction". Managerial Auditing Journal. Volume 20, Issue 4, pp. 394-407.

Formal Ontologies Meet Industry
S. Borgo and L. Lesmo (Eds.)
IOS Press, 2008

Using Agility in Ontology Construction

Amir Azim SHARIFLOO [a,1], Mehrnoush SHAMSFARD [a]
[a] *Natural Language Processing Laboratory, Shahid Beheshti University, Tehran, Iran*

Abstract. Ontology development method is a method that shows how to construct an ontology. Many approaches have been reported for developing ontologies. Almost all of them are derived from traditional software development methods and consequently, have inherited their weaknesses. In the last decade, agile methods have appeared in software development area. These methods attempt to solve some problems of traditional methods. In this paper, one famous agile method that is known as Extreme Programming (XP) is adapted for ontology construction. Also, we report the ontology constructed using this method.

Keywords. Ontology Construction, Agility, Agile Methodology, Extreme Programming

Introduction

Ontology is an explicit, formal specification of a shared conceptualization [1] related to a domain of interest, where formal implies that the ontology must be machine-readable and shared means it should be accepted by a group or community. Many approaches have been reported to develop ontologies such as Uschold and King's proposal [3,4], METHONTOLOGY [6,7], and On-To-Knowledge [8]. Developing ontology is similar to software development from many aspects. Moreover, we can see that ontology and software development methods are affected from similar environmental variables such as customers, changing in requirements, budget, and time.

In recent years, agile methods have appeared in software development area. These methods attempt to solve the problems of traditional methods. Traditional software development methods assume that most of the requirements can be anticipated at the beginning of projects and will remain stable. The inability to evolve the software to be fit with variation of requirements means being unable to business conditions, which leads to business failure [2]. Agile methods such as XP, Crystal methods, and Scrum have been proposed to encounter with today's variable business environment [9]. The goal of agile methods is to allow an organization to be able to deliver quickly and change quickly [10].

However, conventional ontology development methods have inherited the weaknesses of traditional software development methods. On the other hand, we need to apply agility advantages for ontology construction. Consequently, we selected XP and adapted it as an agile method for ontologies and in this paper; we want to propose it. Also, we have applied it for developing an ontology for CMMI-ACQ [11] which is used to improve the processes of acquisition organizations. The rest of the paper is organized as

[1] Corresponding Author: Natural Language Processing Laboratory, Shahid Beheshti University, Tehran, Iran; E-mail: a.shariflоo@mail.sbu.ac.ir.

follows. Section 1 provides motivation of adapting XP for ontology construction. Some background information in ontology engineering is presented in section 2. Section 3 we look at XP and describe the adapted XP method which we call XPOD. Finally, the ontology constructed using this method is explained in section 4.

1. Motivation

Ontology development process is so similar to software development process. This similarity is derived from the similarity between the inputs and outputs of the both. Their inputs are a set of requirements and constraints and their output is an electronic code. In software engineering area, the requirements are some operations that the customer and end users expect to be done by the software system. In fact, the software system is the major output of the process. On the other hand, in ontology engineering, the customer expresses his purpose of constructing the ontology. It could be described in a set of questions. Considering this set, the ontology developers focus on constructing an ontology that has enough abilities to respond to the needed knowledge. Almost, all of the challenges in software engineering such as handling change requests, time to market and the concerns about development team are involved in ontology development. Section 3.3 discusses about the similarities and differences between software and ontology.

Most of traditional approaches in software development such as waterfall, spiral, RUP [14] follow the development process based on the developed plans which are produced in the preliminary stages of the project. Generally, having plans is a good idea and guide project team to develop the product disciplinarily. However there are some limitations. First limitation is the inherent complexity of software [15] and lack of a good requirement elicitation which is related to customers. They do not usually know what they want as a software exactly. Thus, after passing the days and constructing a large amount of the system, they tell some notes about the system that warns to the developers 'the developed system is very different from the desired one'.

Second limitation is the different ranges of the developed documents which imposes an overhead on the team. Updating these documents after any change is another problem. Sometime, some roles of development processes are responsible for producing and updating documents. Even some team members might think about the documents as the product. Whereas the customer only considers the executable software and usually do not pay attention in explanatory and comprehensive documents. Meanwhile, team could not obtain efficient usage of people skill and creativity which is the result of long and rigid plans. In this situation, the relationships among development team are limited to some documents and formal sessions which are not sufficient for developing an acceptable software.

Since the last years of 20th century, agile methods have been introduced as a new generation of methods to develop software systems and they became popular very soon. Indeed, agile methods addressed the problems of traditional development approaches which were frustrating development teams. In 2001, agile manifesto was signed by some leaders of agile community which values some items [16]. It emphasizes on individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation and responding to change over following a plan. Concentrating on these items, we can see that the manifesto has addressed the limitations of traditional development processes.

Regarding the similarity between software and ontology development, the lack of a suitable agile method in ontology engineering area is sensible. An agile method for constructing ontologies almost could bring all of the mentioned benefits in software development. To reach this purpose (having an agile ontology development) we decided to adapt one of the effective and currently used agile methods to employ for ontologies. Therefore, we choose XP as an appropriate candidate which presents some useful practices such as pair programming that can be matched with ontology team roles effectively.

2. Ontology Engineering

Gómez-Pérez defines Ontological Engineering in [17] 'Ontological engineering refers to the set of activities that concern the ontology development process, the ontology life cycle, and the methodologies, tools and languages for building ontologies.' Regarding this definition, for ontology construction we need to consider to ontology development methods, tools and languages. Therefore, in the following, section 3.1 focuses on ontology and some ontology tools and languages. Three reported ontology development methods are briefly described in section 3.2 and section 3.3 compares ontology and software.

2.1. Ontology

An ontology provides a common vocabulary for researchers who require to share information in a domain. It encompasses machine-interpretable definitions of the concepts in the domain and relations among them [18]. The content of a required ontology depends on its applications. It may be very detailed and very limited to a specific domain. It may be involved only major concepts of a large field. It could be full of different restriction rules or without any rules and only a taxonomy of concepts. In general, the ontology requirements make its shape.

Ontologies are being employed in many scientific areas such as Semantic Web, e-Learning, Bioinformatics, and Software Engineering [17]. Some typical purposes of using ontologies are [18]: 1) To share common understanding of information among people or software agents. 2) To enable reuse of domain knowledge. 3) To make domain assumptions explicit 4) To separate domain knowledge from the operational knowledge. 5) To analyze domain knowledge.

One of the important decisions in developing ontologies is to select the language. A various range of ontology formalization languages have been presented such as KIF, OWL, and FLogic. A proper language for formalization of ontology depends on the purpose which has triggered the ontology construction. In addition to, developers usually use ontology tools to decrease the efforts of development. Protégé [12] and Ontolingua [13] are two famous ontology tools.

2.2. Ontology Development Methods

This section presents three well known methods to develop ontologies. We will provide a brief description of each method.

2.2.1. Uschold and King's method

This method was the first method for constructing ontologies [3,4]. To construct an ontology according to that, four following steps must be done:

- Identifying the purpose of ontology.
- Building the ontology that is divided into ontology capture, coding and integrating existing ontologies. Ontology capture is the identification of the key concepts and relationships in the domain of interest. Coding involves explicitly representing the acquired knowledge in a formal language, and finally in integrating, existing ontologies will be integrated.
- Evaluating the ontology.
- Documenting the ontology.

2.2.2. Grüninger and Fox's methodology

This method is based on the experience in developing the TOVE project ontology [5]. It involves theses steps:

- Capturing motivating scenarios.
- Writing informal competency questions.
- Specifying the terminology of the ontology within a formal language.
- Formulating formal competency questions using the terminology of the ontology.
- Specifying axioms and definitions for the terms in the ontology within the formal language.
- Establish conditions for characterizing the completeness of the ontology.

2.2.3. METHONTOLOGY

This method was developed within the Laboratory of Artificial Intelligence at Universidad Politécnica de Madrid. The METHONTOLOGY [6,7] enables the construction of ontologies at the knowledge level. The life cycle of METHONTOLOGY is involved: 1) Specification, 2) Knowledge Acquisition, 3) Integration, 4) Evaluation, 5) Documentation. The life cycle of an ontology is based on the refinement of a prototype. Also, configuration management is carried out simultaneously with the development activities.

2.3. Software and Ontology

What we call 'software' is a machine-readable and executable code. Indeed, all of the activities along software development will result in this code. The code is read by a machine and executed by it. On the other hand, ontology is a machine-readable code that represents knowledge about some general or specific field and might be used by an application. Both software and ontology are coded by formal languages. As we can see, both of them are composed of code that is developed for responding to the development purpose. But the major difference between software code and ontology code is the execution. Software code is executable whereas ontology code is a formal representation of knowledge and could be applied by softwares which are executable. Knowledge based systems are one instance of this kind of softwares.

The similarity in nature causes some similarities in development process of softwares and ontologies. We can find a mapping between the building procedures of them.

Table 1. A mapping between typical activities of ontology engineering and software engineering.

Ontology Engineering	Software Engineering
Problem Definition	Problem Definition
Knowledge Acquisition	Analysis
Conceptualization	Design
Formalization	Implementation
Evaluation	Test
Maintenance	Maintenance

Table 1 provides this mapping between typical steps of ontology engineering and software engineering.

As is shown in Table 1, in the building procedure of the both we have to define the problem which must be solved. Problem Definition determines the purpose of ontology and some constraints about it. Without a clear purpose of ontology, the ontology development will be encountered with different problems. During Knowledge Acquisition process, the requirements of ontology are illustrated. According to these requirements, the acquisition process of the needed knowledge begins. The requirements of ontology usually can be described by questions. Considering the acquired knowledge, the ontology is designed. Design activity determines the structure of ontology and we call it Conceptualization. 'Which concepts must be involved in ontology?' and 'Which relations are required?' are some basic questions that appear and conceptualization process attempts to satisfy them. Sometimes, conceptualization is known as informal ontology modeling and it could be done using graphical models such as Semantic Nets. The next step is Formalization and it is coding the ontology by a formal language. This step is comparable with software implementation. The models made in conceptualization are applied in formalization. It is necessary to evaluate the constructed ontology to find out its errors or shortcomings. This process is called ontology Evaluation. Finally, due to the fact that knowledge is changing in the world, every developed ontology has to be maintained and this process is called Maintenance.

3. Toward agile ontology development

Before presenting the proposed method, we need to review XP because it is the foundation of the method. Section 4.1 look at XP's values, practices, roles and process model in brief. After that in section 4.2, we describe XPOD as an adapted XP in ontology engineering area.

3.1. A look in Extreme Programming

In 1999, Kent Beck wrote his famous book [2] and presented XP as a lightweight method which was became one of the most popular during few years. The introduction of XP has been widely acknowledged as the starting point for the various agile software development approaches [9]. XP is based on four values: Communication, Simplicity, Feedback, and Courage. Considering them, some principles and 12 practices are defined that through a process model create XP's basis. The practices of XP [2] are:

- Planning Game. 'Quickly determine the scope of the next release by combining business priorities and technical estimates.' Beck says [2]. During this practice, customer and programmers interact to each other to determine the requirements as story cards and estimate the effort needed for their implementation. As a result, the scope and timing of releases are defined.
- Small Releases. Software is not constructed all in once as a Big Bang. Ordinarily, it is developed as several releases and through them software are developed incrementally with small functionality added. Each release should be as small as possible, including the most valuable business requirements. The release has to make sense as a whole [2].
- Simple Design. The emphasis of XP is on designing the system as simple as possible. Thus, pair programmers choose the simplest solution. The complexity which is not necessary is removed immediately. This practice is derived from simplicity value, clearly.
- Testing. Before coding each part of system, pair programmers write the related test cases at first. The implemented part is test by them. Because of that, XP is called a test-driven development method.
- Metaphor. It describes how the system works. The description obtains by some stories about the systems that express the system functionality to the customer and development team.
- Refactoring. During refactoring, the system is restructured by removing duplication, and simplifying the complex structures.
- Pair Programming. Due to this practice, two programmers collaboratively write the code at one computer. One of them focuses on the design and test issues and another programmer codes.
- Collective Ownership. Any member of the XP team is allowed to change any piece of the code at any time.
- Continuous Integration. A new piece of code is integrated into the code-base as soon as it is ready. Thus, the system is integrated and built many times a day. All tests are run and they have to be passed for the changes in the code to be accepted.
- 40-Hour week. The maximum number of working hours in a week is 40.
- On-site Customer. At least, a customer has to participate in development team and help the team in eliciting the requirements.
- Coding Standards. Coding standards must been followed by the programmers. It helps the team in understanding the code and having a good communication around the team.

XP process model is shown in Figure 1. In Exploration phase, the customers write out the story cards that they wish to be included in the first release. Each story card describes a feature to be added into the software. Planning phase sets the priority order for the stories and an agreement of the contents of the first small release is made. (Iterations to Release) Productionizing phase requires extra testing and checking of the performance before the software can be released to the customer. In Maintenance phase, customer requests for adding new features to software or changing some parts. Death phase is near when the customer does no longer have any feature to be implemented.

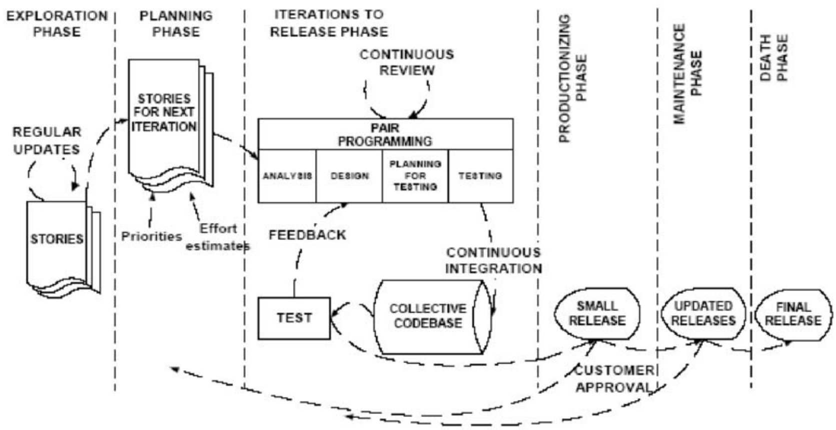

Figure 1. The process model of XP [9].

3.2. XPOD

Since each software project has own characteristics, it needs a development method that can be adapted to that. Ordinarily, developers select a known development process and tailor it considering the project characteristics. This tailoring could lead to some changes such as removing some roles or changing the process model. XPOD could be useful for the development team which needs to deliver ontologies quickly and satisfy the customer efficiently.

To employ XP as an ontology construction method, we need to map the software space to ontology space which we discussed it in section 3.3. XP's values are: 1) Communication 2) Simplicity 3) Feedback 4) Courage. XPOD has inherited them completely. However from a black box perspective, the ontology development through XPOD is very similar to XP because it has a similar process model and practices. But the difference is in their details. XPOD process model is shown in figure 2. In Exploration phase, the customer writes out the story cards that they wish to be included in the first release. Each story card describes an important competency question that s/he expects to be answered by the ontology. Using these competency questions, the ontology scope is defined. Also, customer makes decision about reusing other ontology. In this circumstance, team makes a spike on the selected ontology for reusing and learns about its abilities and shortcomings.

Planning phase sets the priority order for the questions and an agreement of the contents of the first small release is made. Based on ontology scope, we define the scope of the first release and split it to sub-domains. Also, Planning involves selecting proper resources required for knowledge acquisition. In this phase, the standards for conceptualization and formalization are established too.

Iterations to release phase includes several iterations of the systems before the first release. In this phase, conceptualization and formalization is done using pair programming that we call it pair modeling in XPOD. Pair modeling is done with two persons, one of them is domain expert and other one is ontologist. At first, test cases are written in question format with proper details. Second, domain expert answers the questions or extracts their answers from resources, and then these answers are conceptualized as

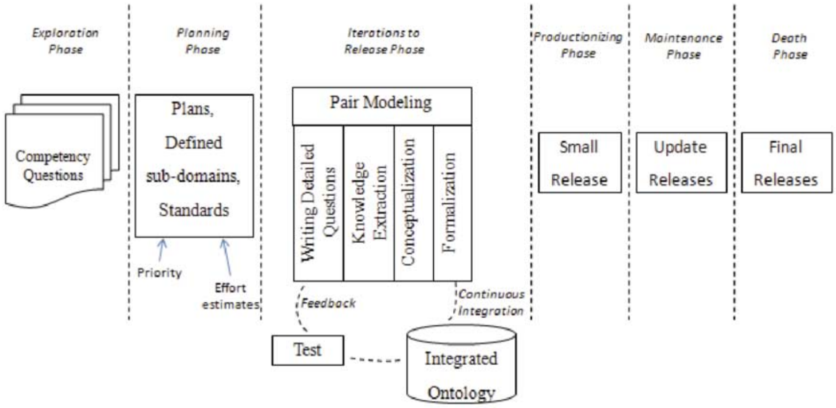

Figure 2. The process model of XPOD.

concepts, relations and rules by domain expert and ontologist collaboratively. Finally, ontologist formalizes them using a formal ontology language. Team uses a configuration management system for maintaining an integrated ontology every time. Therefore, changes on ontology by every pair are shown for other pairs. At the end of each iteration, global evaluation is done on all of the ontology to control the ontology consistency and completeness. Productionizing phase requires extra testing and checking of the quality of the ontology before it can be released to the customer. In Maintenance phase, customer requests to add new knowledge to ontology or change some parts. Death phase is near when the customer does no longer have any feature to be formalized.

In the following, we concentrate on the effected practices.

- Planning Game. Ontologists, domain experts and customer collaboratively work to determine the scope of releases. They estimate the effort needed for the construction of customer competency questions and the customer then decides about the scope and timing of releases. Domain experts actively participate in this practice because they know the space complexity of the domain.
- Small Releases. Ontology is developed incrementally with sub domains added.
- Metaphor. Ontology is presented graphically for all team members and they communicate on this graphical schema. It seems good to have a whiteboard which presents the skeleton (major concepts and relations) of the ontology. This graphical schema is similar to Metaphor in XP.
- Refactoring. Restructuring the ontology by removing simplifying, duplication and solving inconsistencies.
- Pair Modeling. Two persons formalize ontology at one computer. One of them is ontologist who is expert in conceptualization and formalization, other person is domain expert that extracts required knowledge for ontology. Both of them work together on common part of ontology.

Collective ownership, continuous integration, simple design, testing, 40-hour week and coding standards are being employed during ontology development. Also, a customer is involved in the project as a member of team. This attendance helps the team in getting feedbacks continuously and efficiently. In compare to XP, XPOD needs a new

role which is domain expert. Domain expert who has sufficient knowledge about a specific domain extracts the knowledge required to satisfy the purpose and requirements of the project. Domain expert complements ontologist's knowledge about the domain. If the domain is a public domain and does not contain so specific information, it will be possible to substitute the domain expert role with another ontologist. Since both of the ontologists will have good knowledge about ontology coding, they close to real pair programming.

4. CMMI-ACQ Ontology[2]

We attempted to apply XPOD in developing an ontology for CMMI-ACQ domain. The project has done successful in two months. In the following, we explain the developing route step by step.

Since the ontology must be developed considering SUMO [19] as a reusable ontology, the team had to understand the concepts and relations of that. Thus first phase of the development was to focus on SUMO. Therefore team did a spike on learning SUMO. Meanwhile, customer who knew the domain of CMMI-ACQ determined the purpose of constructing ontology and wrote essential competency questions.

During second phase, considering the most important concepts of the CMMI-ACQ domain were identified in addition to their relations among them. This step was a critical and significant step, because through that, the ontology boundary was determined by the ontology scope. This scope consists of main concepts and relations. The main concepts and the relations among them are shown in figure 3. Considering that, the project manager of development team decomposed the domains into 23 sub domains that 22 of them cover 22 process areas of CMMI-ACQ and one sub domain covers its generic goals and generic practices.

Through Iterations to Release phase, the sub domains were distributed among pair modeling teams and the ontology was constructed. In each pair modeling team, one person was a domain expert who explained and conceptualized the domain concepts and relations. The other person was an ontologist who formalized the developed conceptual models in the SUO-KIF language, based on defined concepts and relations in SUMO. He controlled the ontology quality too. When each sub domain was fully completed, it was evaluated from covering aspects of the typical questions and the final version would be delivered to the project manager who was also an ontologist. He reviewed the developed ontology in collaboration with a domain expert of CMMI-ACQ, and integrated them as a whole ontology.

During this step, the team employed a unique table including all the concepts and relations which were used to maintain the consistency of final ontology. Another advantage of this table is to increase the reusability of the defined concepts by other couples and as a result, a kind of collaboration among couples is created. Also, the team employed a whiteboard as metaphor that showed the main concepts and relations which expressed the boundary (Figure 3). Any change in the boundary of the ontology was applied in the mentioned whiteboard by the project manager. At the final step, the ontology as a whole

[2] This ontology won SUMO Prize 2007. It is available in: (`http://labs.sbu.ac.ir/nlp/extra/CMMIACQ.kif`)

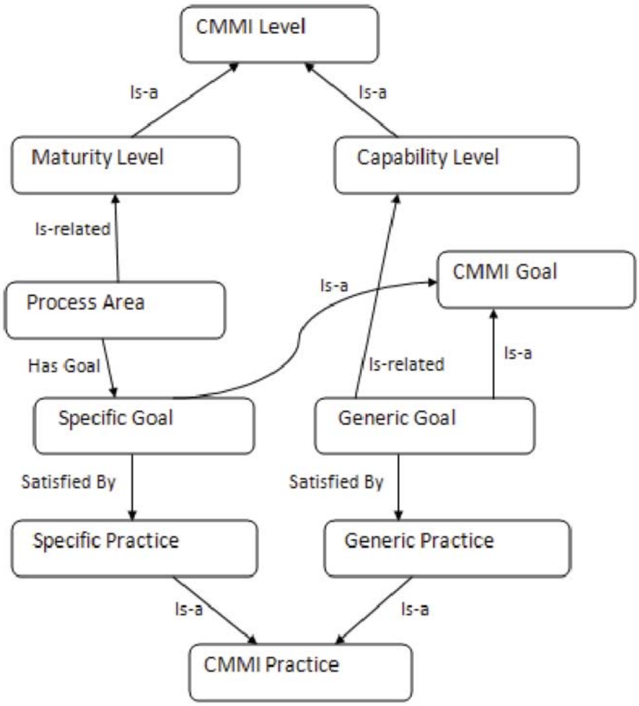

Figure 3. CMMI-ACQ main concepts and relations among them.

was tested by domain expert questions. Each identified problem was introduced as a new task and assigned to a pair.

The main purpose of developing the CMMI-ACQ ontology is using it into a tool which assesses maturity level of the organization automatically, and helps improving its processes. The tool interfaces get the data about the performing state of generic and specific practices around the organization from user. A part of this tool codes the gathered data into a SUO-KIF file. After attaching this file to the knowledge base of the tool, the tool reports the current state of the organization including related maturity level and capability level of each process areas, etc. At this point, user indicates a desirable state which could be some specific level for some process area or simply the desirable maturity level. Tool finds the gap with comparing the current and desirable state, afterward presents a list of required practices to be done. This process would be very time consuming and error-prone if it was performed manually.

5. Conclusion

In this paper, we introduced XPOD, a methodology for developing ontologies, which is based on Extreme Programming, the most popular agile software development method. However XPOD like other agile methods can bring some benefits that are important for each ontology development team. Also, we explained XPOD‘s steps around a real project. During the project, we developed an ontology for CMMI-ACQ model which won SUMO Prize 2007.

References

[1] Borst, W.: Construction of Engineering Ontologies. In: Centre for Telematica and Information Technology, University of Tweenty. Enschede, The Netherlands (1997)

[2] Beck, K., Andres, C.: Extreme Programming Explained-Embrace Change. Addison-Wesley, (1999)

[3] Uschold, M.: Building Ontologies: Towards A Unified Methodology. In: Proc. Expert Systems 96, Cambridge, December 16-18th (1996)

[4] Uschold, M. and KING, M.: Towards A Methodology for Building Ontologies. In: IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Canada (1995)

[5] GRÜNINGER, M. and FOX, M.S.: Methodology for the Design and Evaluation of Ontologies. In: IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, August 19-20th (1995)

[6] FERNANDEZ, M., GOMEZ-PEREZ, A. and JURISTO, N.: METHONTOLOGY: From Ontological Art Towards Ontological Engineering. In: AAAI-97 Spring Symposium on Ontological Engineering, Stanford University, March 24-26th (1997)

[7] GOMEZ-PEREZ, A., FERNANDEZ, M. and DE VICENTE, A.J.: Towards a Method to Conceptualize Domain Ontologies. In: ECAI-96 Workshop on Ontological engineering, Budapest (1996)

[8] On-To-Knowledge Methodology: www.ontoknowledge.org/downl/del18.pdf

[9] Abrahamsson, P., Salo O., Ronkainen, J., Warsta, J.: Agile software development methods. VTT Electronics (2002)

[10] Highsmith J.: Agile Software Development Ecosystems. Addison Wesley (2002)

[11] Dodson K. M., Hofmann H.F., Ramani G.S., Yedlin D.K.,: Adapting CMMI for acquisition Organizations - A Preliminary Report. Software Engineering Institute, Carnegie Mellon university, (2006)

[12] Protégé: http://protege.stanford.edu/

[13] Ontolingua: www.ksl.stanford.edu/software/ontolingua/

[14] Kruchten, P.: Rational Unified Process, The: An Introduction, Third Edition. Addison Wesley (2003)

[15] Booch, G.: Object-Oriented Analysis and Design with Applications. Addison Wesley (1993)

[16] Agile Manifesto: www.agilemanifesto.org

[17] Gómez-Pérez, A., Fernández-López, M., Corcho, O.: Ontology Engineering. Springer (2004)

[18] Ontology Development 101: A Guide to Creating Your First Ontology: www-ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness-abstract.html

[19] Suggested Upper Ontology: www.ontologyportal.org

Formal Ontologies Meet Industry
S. Borgo and L. Lesmo (Eds.)
IOS Press, 2008

IPAS ontology development

David W. FOWLER, Quentin REUL and Derek SLEEMAN [1]
University of Aberdeen, Aberdeen, United Kingdom

Abstract. There is a trend in some manufacturing industries to move from selling products to providing services. As a result, designers must consider the life-cycle costs. In the aero industry, for example, this must be considered as well as weight, performance and manufacturing cost. The IPAS (Integrated Products and Services) project is intended to utilise Semantic Web technologies in order to provide feedback of information/knowledge acquired during operation of a product to the product's designers, and also to reuse knowledge from previous product designs. As part of IPAS, ontologies describing the products and processes have been created in order to allow service knowledge to be represented and shared. The design and implementation of these ontologies, and their planned future evolution, is described here. We also try to draw some lessons from our experiences.

Keywords. Ontologies, IPAS, design patterns

Introduction

In the past, manufacturers have designed products, manufactured them, and sold them to end users. There is a trend now away from simply providing products to including maintenance services. A good example of this is the Rolls-Royce TotalCare® package [1], where Rolls-Royce undertakes to provide flying hours to the customer at a fixed cost per hour. As the variable costs of operating the products is borne by the manufacturer (while the end user pays a fixed price), there is an added incentive to design the product in order to reduce overall lifecycle cost. To do this, the designers require to access knowledge from previous product service experience. This knowledge is contained in many different locations and types of documents. Additionally, we are concerned with making that information available on workstations attached to the company's intranet, and so we are exploring the use of Semantic Web technologies in the delivery of these services.

IPAS (Integrated Products and Services)[2] is a three year project co-funded by the UK Technology Strategy Board's Collaborative Research and Development programme[3] and Rolls-Royce plc, that involves researchers from ten universities and companies, with specialisms in artificial intelligence, engineering, and organisational psychology. One of the principal aims of IPAS is to provide feedback from in-service operation of products to the design of new products and services. IPAS covers fields such as:

[1] Corresponding Author: Computing Science, University of Aberdeen, Aberdeen AB24 3UE, United Kingdom; E-mail: d.sleeman@abdn.ac.uk.

[2] http://www.3worlds.org/

[3] http://www.berr.gov.uk/dius/innovation/technologystrategyboard/index.html

- Knowledge representation;
- Extraction of information from textual documents;
- Life cycle cost modelling;
- Process analysis;
- Social network analysis.

Within IPAS, we have mainly used ontologies to support the first two of these areas as part of Semantic Web-based demonstrators [2]. The intention of the project is that it will produce a specification for an integrated system to be fully implemented in the future.

The rest of the paper is organised as follows. Section 1 describes related work on ontology design, evolution and modularisation. Section 2 explains how the initial version of the IPAS ontologies was created. Section 3 describes the shortcomings of the ontologies, how they were redesigned in the light of this feedback, the principles that were used, and the ontology-based applications that have been developed in IPAS. In Section 4 we describe the remaining issues and further work, including plans for future evolution of the ontologies. Finally, in Section 5 we attempt to draw some conclusions.

Note on terminology. In this paper, to describe the ontologies developed during IPAS, the plural term *ontologies* has been used, whether they comprise one OWL file or several. These comprise several domain ontologies, covering engines, modules, materials, and so on. Each of these is referred to as *an ontology*. The only difficulty with using this terminology is that the initial version of the IPAS ontologies was held in one OWL file, and was previously referred to as *the IPAS ontology*. However, for consistency, and as it did cover more than one domain, in this paper we use the term *IPAS ontologies* for all of the versions. Furthermore, we use the term *part* to refer to all hardware (including engines and modules), and the term *component* for all parts smaller than a module.

1. Background

In this section we describe related work on how ontologies can be designed. Gruninger and Fox [3] propose an approach to engineering ontologies based on four steps. First the knowledge engineer defines the questions that the ontology must be able to answer. Secondly the vocabulary of the ontology is defined, i.e. its classes and properties. The third step is to use the vocabulary defined in the previous step to specify the definitions and constraints. Finally, the engineer tests the competency of the ontology by attempting to answer the questions defined in the first step.

Similarly, Noy and McGuiness [4] propose a methodology with the following steps (some terms have been updated to reflect OWL usage):

1. Determine the domain and scope of the ontology.
2. Consider reusing existing ontologies.
3. Enumerate important terms in the ontology.
4. Define the classes and the class hierarchy.
5. Define the properties of classes.
6. Define the restrictions on properties.
7. Create instances.

Noy and McGuinness stress that “ontology development is necessarily an iterative process”, and that as an ontology is a model of reality, it will have to be revised and debugged. This must happen when: (a) it is found that the ontology cannot answer an existing question or support an activity, (b) it is found that the ontology is inconsistent (some classes cannot have instances, due to logical contradictions), (c) the existing ontology appears to be correct, but could be restructured in a more logical fashion, (d) new competency questions are added which the ontology does not currently address, or (e) there are major changes to the company’s data model or business model.

Lopez et al. [5] propose a methodology, called Methontology, to provide a user-friendly approach to knowledge acquisition by non-knowledge engineers, and an effective method to domain-knowledge-model construction and validation without specifying a particular representational schema. The process comprises five steps, namely specification, conceptualisation, knowledge acquisition, integration and implementation. The Ontology Design Environment (ODE) performs the latter step.

2. Initial IPAS ontologies design

As the ontologies were required to extract data from text documents, and then to support working demonstrators (see Section 3.3 for details of these), it was decided to create an initial version of the ontologies quickly. This was then followed by a stage in which the ontologies were refined in the light of experience, but subsequently further versions of the ontologies were produced as the scope of the project expanded. Therefore, the initial version (v1.0) was planned for three months into the project, with a second version (v2.0) at nine months, v3.0 at fifteen months, and the final version, v4.0, two years after the project start. This meant that other partners in IPAS would be able to start developing their subsystems fairly quickly, with the understanding that early versions of the ontologies would be small, and subject to change.

It is important to realise that the scope of the ontologies was not really clear from the start, but was modified over time, as it became clearer what the software demonstrators were to do, and what activities they would support. The main activity supported by the demonstrators is to provide feedback from the operation of existing engines, and so we concentrated on representing existing engines. While for future systems it might be useful to represent future designs (consisting of abstract potential components, rather than existing ones), it has not been done in the current ontologies.

We chose OWL [6] as the language for ontology development because of the wide range of software tools that can be used with it, due to our previous experience with it, and also because it is an open, nonproprietary, standard (a W3C Recommendation). Also, as it was not clear from the start of the project exactly what the knowledge was to be represented, we could not decide whether OWL was fully adequate for the task, but we were encouraged by the fact that other (proprietary) knowledge representation efforts for engineering domains have also used OWL, e.g. PLCS [7] and ISO-15926 [8]. OWL has three sublanguages: OWL Lite, OWL DL and OWL Full, varying in expressiveness and tractability for reasoning (OWL Lite being the least expressive, but most tractable, and OWL Full being the most expressive, but least tractable). It was decided to use OWL DL as a tradeoff between expressiveness and tractability; this decision had important implications later. The main one is that often it is required to refer to a class rather than

to an individual (i.e. as the value of a property). OWL Full allows reference to classes, whereas OWL DL does not.

The initial versions of the ontologies were intended to cover the event reports (describing incidents that led to maintenance actions being taken) that were to be fed back to the designers, and also the strategy sheets for components (plans for mitigating deterioration mechanisms for components). To ensure that the ontologies could represent the principal concepts in these documents, some sample reports were examined, and the key concepts extracted manually. Our intended approach for developing ontologies largely followed the approach of Noy and McGuinness [4] (although this predates the introduction of OWL), supplemented with insights from other sources such as [9] which is specifically about OWL (and recommends an approach that allows for modularisation, although doesn't give details as to how this can be done). It is fair to say that we did not fully appreciate at the start of the project the importance of modularity in ontologies, and the early versions of the ontologies were monolithic, with everything stored in one OWL file. This caused problems that had to be addressed later (see Section 3.1).

It was clear that it would be necessary to represent engines and their components. There are several distinct types of entity that could be represented, including: (a) actual physical parts, i.e. those that have been manufactured and exist as distinct individuals, (b) specific, existing, *types* of part (used in cases where it is necessary to refer to a type of part, but not a specific individual part, and (c) a more generic type of part, not necessarily existing (perhaps a part that is being designed). For the ontologies so far, we have concentrated on the first two of these types (namely, existing parts). Version 1.0 of the ontologies included a very small, hand-built, ontology covering a few parts only, but for more extensive uses (including the subsequent demonstrators), many more part types would be required. To build this ontology from scratch would have been prohibitively expensive (the current version of the ontologies has around 3000 types of part, covering nine engine types). Rolls-Royce provided a taxonomy in the form of a series of linked HTML pages that had been developed for a previous project, and contained a model of an engine.

This taxonomy covered a wide range (e.g. parts, processes, materials, etc.), but there were several problems with it. Firstly, different types of relationship were represented with the same taxonomic relation. For example, in the engine section, the links were part-of; in the materials section the links were is-a; in the process section they were subprocesses (or temporal part-of). Secondly, the engine section of the taxonomy was generic, covering a typical engine rather than any specific engine(s). This may be sufficient for some purposes, but not for applications where comparisons are made between engine types. Finally, there is no further documentation associated with any of the nodes in the taxonomy. Therefore, the only explicit information is the names of the nodes, and the links between them (and the exact nature of these links has to be inferred). However, as it was the best information source available at the time, and machine-readable, the section of the taxonomy dealing with the engine was used as part of the early ontologies, with the taxonomic links being replaced by part-of relations. It should be emphasised that we were not representing a complete engine design with a complete breakdown of parts, as this would require a means of representing multiple instances of the same part being part of an assembly. Instead, we used an "is-allowed-to-be-part-of" relationship only.

Another major section of the ontologies covered deterioration mechanisms. These are the processes by which components deteriorate, and eventually are either replaced

after failing an inspection, or fail to perform their task. Deterioration mechanisms are clearly important in IPAS, as the causes of failures or replacements are to be recorded, and fed back to designers. The initial deterioration mechanism ontology was created by referring to a *Mechanism Prompt List* used at Rolls-Royce. The Mechanism Prompt List was intended to remind designers of various possible mechanisms while considering a new design, and comprises main categories (e.g. Fatigue, Creep, Adhesive wear) and subcategories (e.g. for Fatigue: Fretting, Corrosive, Rolling Contact, etc.). The Mechanism Prompt List was used to construct the deterioration mechanism ontology fairly straightforwardly. The main categories were used as concepts in the first level of the class hierarchy. The subcategories were used as subclasses of the main classes. Some minor clarifications were carried out by referring to engineering textbooks, and by discussing the class hierarchy with design engineers.

3. Re-engineering the ontologies

After the initial ontology development had been completed, time was allocated to (a) get feedback from other partners, (b) to reflect on the design of the ontologies, and (c) to gain access to more documentation to help in enhancing the ontologies. Feedback from the other partners was difficult to get, probably due to a steep learning curve in using Protégé and OWL for many. Automatically produced HTML documentation for the ontologies (OWLDoc [10]) and graphical representations of the class hierarchy of the ontologies were used to try to overcome this problem, but with limited success. Most feedback came later in the project, when the demonstrators were implemented (see Section 3.3 for details). However, the ontologies were enhanced by reviewing their design, and attempting to restructure them by using design patterns [11,12] and by reflecting on best practices. Design patterns allow the use of good solutions to common ontology design problems to be reused (the solution may be more efficient, conceptually elegant, or allow reasoning to be performed). Also, gaining access to more documents provided resources to allow the ontologies to be enlarged.

The conclusions drawn immediately were firstly that the ontologies would have to be split into modules for them to be manageable as they grew, and secondly that ontology design issues would have to be examined carefully. Examples of such design issues were most obvious in representing engines and components (discussed below). The initial ontologies were not capable of such representations, and had to be redesigned. Following the appraisal of the initial version of the ontologies, further versions were produced, using better design practices and the additional sources of knowledge that had become available.

3.1. Ontology modularisation and evolution

The initial version of the ontologies was monolithic, in that everything was contained in a single OWL file. This was not a major problem while the ontologies were small, but when the engine ontology was added, it became clear that the single ontology file would have to be separated into several smaller domain ontologies. This process was repeated during the evolution of the ontologies, with the engine ontology being divided into separate engines, modules and parts ontologies. Also, ontologies describing different

Table 1. The evolution of the IPAS ontologies

	v1.0	v2.0	v3.0	v4.2
Classes	72	452	375	7863
Object properties	30	36	15	78
Datatype properties	34	60	23	124
Files	1	2	3	14

types of reports were subdivided. This modularisation had the advantages that individual ontologies could be worked on independently and reused.

Table 1 gives an overview of how the ontologies have changed since the beginning of the project. The drop in size between v2.0 and v3.0 is due to the ontologies being radically overhauled, due to (a) a redesign using ontology design patterns, and (b) a better understanding of what the ontology should describe (the "early prototype" approach to building ontologies in this project had the advantage of producing early versions for inspection and use, but the disadvantage that the scope of the ontologies was not fully understood while the ontologies were being built) . There was a large increase in size between v3.2 and v3.3 as new sources of component data became available and automated methods were developed to extract this data.

One of the ontology design patterns that was most frequently used in building the IPAS ontologies is the use of classes as property values. This construct is used (a) to refer to a generic (or typical) engine or part rather than a specific physical object, (b) to refer to deterioration mechanisms or materials, which logically seem to be better represented by classes than individuals, and (c) to refer to specific parts that are not "trackable" (they do not have a serial number, so it is not possible to distinguish subsequently between two individuals of the same class . In [13], several possible approaches to work around the OWL DL restriction of not being able to use classes directly as property values are given. The second of these approaches was chosen, which involves creating a "dummy" individual for each class that may need to be referred to. The main reason for the choice was simplicity; this may be reviewed at a later stage. To achieve the effect of using a class as a property value, all that needs to be done is to refer to the dummy individual instead. This also has the additional benefit of allowing assertions about classes. In OWL, classes can be in the domains of special kinds of properties, called *annotation properties*, but these cannot have restrictions placed upon them (so, for example, the type or allowed values of an annotation property's range cannot be specified). By using a dummy individual to stand for the class, the properties of the class can be represented as the properties of the individual.

Representing the materials that parts are made of provides an example of how the approach was used in the IPAS ontologies. In an earlier version of the parts ontology, the material that a generic part is made of was indicated by an OWL annotation property (`has_material_code`). The class for the part type has this property, and its range is a string (the three letter material code). See Figure 1 for a representation of this approach. This was clearly not ideal, but this is forced upon us as classes can only have annotation properties, which cannot have range restrictions. To find the material in the materials ontology, a search would have to be performed on the material code. The current design revised the parts ontology so that each part class has a dummy individual associated with it, and uses a `has_material` property to link this class to an individual from the

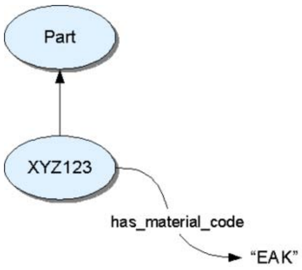

Figure 1. An early ontology design to represent the material of a component using annotation properties

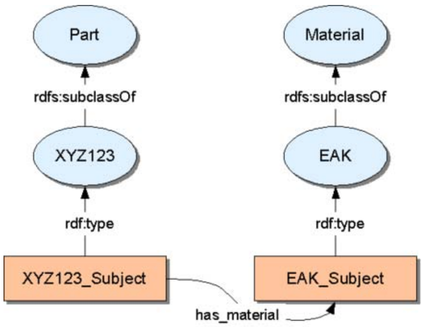

Figure 2. A more sophisticated design using classes as property values

materials ontology (Figure 2). In this, and in other ontologies, the dummy individuals were added by processing the OWL-DL ontology with a simple Java program.

As an example of modularisation, semi-automated ontology construction, and using classes as property values, we consider the development of the ontology describing engines, modules, and their components. The initial ontology was divided into three sections:

Engines the top level entities;
Modules one of a small number of interchangeable parts in an engine;
Components anything below the level of a module, including indivisible parts, and assemblies.

The engine is the highest level entity that we are concerned with in the parts ontologies. We have currently dealt with versions of the RB211 Trent engine.

The Trent engine has eight main modules. The modules ontology has one class and one dummy individual for each of these module types — we are currently not aiming to represent every subvariation of module that exists.

The components ontology was built up using parts lists supplied in comma separated value (CSV) format. Each row contained an entry for a particular part (including parts that were versions of older parts). The details for each part type includes:

ATA location This is a multi-field entry that identifies the particular part on a drawing;
Part number The company's code number for all parts of this type;
Category A high level description, such as "VANE", "BOLT", etc. This field was limited to eight characters;
Subcategory A further description — "BIHEX HD" is an example of a subcategory for the category "BOLT". Around 50% of the entries had a subcategory entry, the rest did not;
Detail description A final text description, often giving dimensions, such as ".190 DIA X .438" for a "BIHEX HD BOLT". Around 40% of entries had an entry for this field;
Material code A three letter code used by the company to identify materials;
Service bulletin number An ID number of a document relating to the part, often describing modifications that were required to update the part from a previous version.

The category and subcategory fields were used to create a shallow class hierarchy for the parts. The part number was used to make a new leaf class in the hierarchy. Dummy individuals for the leaf classes were created, and the other fields in the spreadsheet were used to fill in the property values for these.

As part of the components ontology, we also represented *features*. Features are the locations on a part, usually with some functional purpose. They are important for our purposes, as deterioration or damage is often described as occurring on a particular feature of a part. We do not currently represent features in detail, instead just the name of the feature is used. For example, we record that a HP NGV has a feature called "trailing edge", but not its size, or relation to other features. We used SKOS [14] to represent the names of features, and their synonyms. Features will be put into a separate module in future versions of the IPAS ontologies.

3.2. Other IPAS ontologies

A materials ontology is used to represent the materials that components are made of, together with some of their physical and economic properties. The ontology imports two other existing ontologies to describe physical units[4] (for the temperature, density, and tensile strength properties), and currencies[5] (for the cost property). The physical units ontology allows for units in different systems (e.g. metric and British Imperial) to be related. For example, `gram` and `pound` are both individuals of `MassUnit`, and properties `hasUnit` (objecttype) and `factor` (datatype) are used to relate units. So, to assert that a pound is equal to 453.59238 grams, the following statements are made:

```
pound hasUnit gram
pound factor 453.59238
```

The deterioration mechanism ontology remained broadly similar to the original version (although it is now a separate module in the IPAS ontologies).

A new ontology describes *mechanism records* — a means of storing information from service records about deterioration mechanisms and their actual occurrences, and/or details of how the mechanism was "designed against" (i.e. how the designers prevented or reduced the occurrence of the mechanism; this is clearly important knowledge to store, as otherwise future designers may unwittingly reintroduce the mechanism). The term *mechanism template* is used to describe the format of the mechanism records — that is, a mechanism template is a class, and the mechanism records are instances of that class. The mechanism template ontology was developed by having engineers describe the types of knowledge that they thought would be required to help designers design future components. For the initial version of the ontology, many of the more complex fields of the mechanism template were left as free text entries (for example, the details of mechanism reduction/prevention is represented as a collection of strategies for inspection, maintainability, repairability, etc., but each of these strategies is free text). In later versions, these may be replaced by more structured alternatives.

---

[4] http://archive.astro.umd.edu/ont/index.html from the Department of Astronomy, University of Maryland

[5] http://www.daml.ecs.soton.ac.uk/ont/currency.owl from the University of Southampton

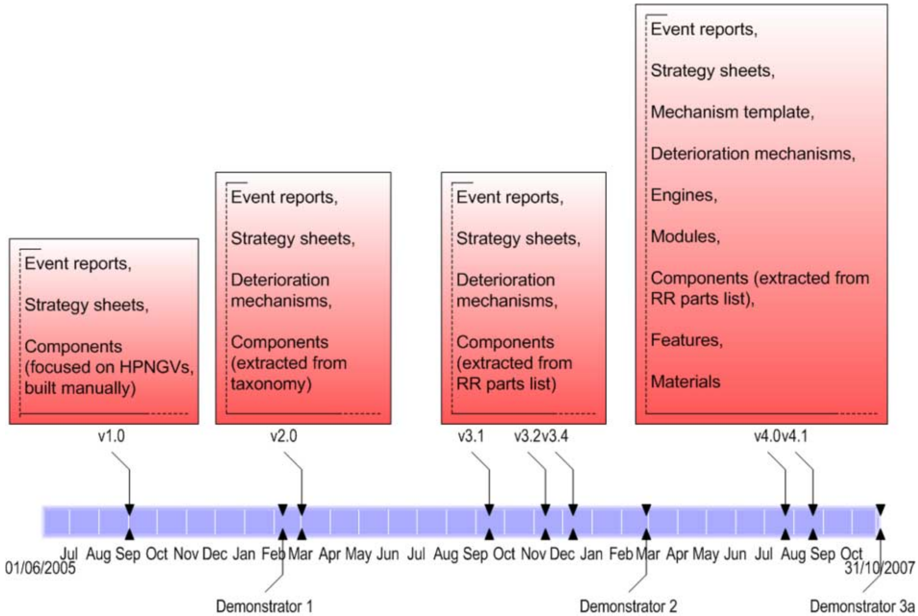

Figure 3. A timeline of the ontology development and the IPAS demonstrators

3.3. Applications of the ontologies

The IPAS project's deliverables include several demonstrators which are intended to illustrate the use of semantic web technology to retrieve service data for use in design. Currently two such demonstrators have been completed; work is in progress on a third. The first demonstrator allowed users to plot graphs from service data, (such as numbers of service events per engine type).

The second demonstrator supported more complex searches for individual service events, and retrieved service events for different deterioration mechanisms. This demonstrator also retrieved images of damaged parts. This demonstrator uses data that was extracted in the form of RDF triples from service event reports, and allows the user to search for reports based on the engine and part types.

The third demonstrator is intended to handle mechanism records, which will enable domain users to find details of previous events, to create new mechanism records, and search for existing ones. The user (a senior engineer, or "knowledge guru") will take a more active role in creating knowledge that other users (e.g. designers) will then be able to search through.

Figure 3 shows a timeline of the IPAS ontology development, together with the demonstrators that were implemented at the time of writing (the third demonstrator is being implemented in three smaller stages: 3a, 3b and 3c).

4. Remaining issues and future work

In this section we describe some outstanding issues concerning the design of the ontologies, and possible approaches to solving them. There are several problems with the

current IPAS ontologies, including: the cleaning of the ontologies that have been built semi-automatically; whether to relate the ontologies to an upper ontology; and how to represent change over time.

A significant disadvantage of the automated method of creating the parts ontology from a spreadsheet is that the spreadsheet contains names which lack consistency, (e.g. naming and terminating conventions have not been followed consistently), and these inconsistencies need to be resolved in some part of the process. For example, the pairs of terms `ADAPTER` and `ADAPTOR`, and `ARRGT` and `ARRNGMNT`, are found in the parts spreadsheet, and presumably refer to the same entities. It would be possible to use string comparison algorithms to suggest likely occurrences of this problem, but the final decision would have to be made by a domain expert. Ideally, checks would be implemented to ensure that these variants cannot occur in the first place. We are investigating an approach using CleOn (formerly called CleanOnto) and SKOS to do this [15].

Currently, there is not much inference being done with the IPAS ontologies — the implemented demonstrators use the ontologies to connect separate data sources. The main inferencing is simple transitive reasoning, whereby if an individual is a member of a class, it can be inferred that it is also a member of all superclasses of that class (this reasoning is handled by the Sesame triplestore). The example of parts having multiple names would be one simple example where inferencing could be used, by stating that two or more classes are equivalent. This is being done in the latest version of the components ontology. Another example would be in classifying events according to severity or importance. We also have to investigate the scalability of any reasoning, as the ontologies are rather large. One approach would be to use approximate reasoning, as in Pan and Thomas [16].

The choice of design patterns may be revisited — the choice of dummy individuals to represent classes has the advantage of simplicity, but may have disadvantages if more reasoning is to be performed.

The use of upper ontologies should be considered. An upper ontology is a high level ontology, not specific to any single domain. It would be possible to use an existing upper ontology that describes high level terms such as `Physical thing`, `Process`, `Person`, etc., that could be used as superconcepts for the IPAS ontologies terms. The advantage of this approach is that other existing domain ontologies that refer to the same upper ontology could be easily integrated with the IPAS ontologies. However, there are several competing upper ontologies available, and there is no commonly agreed standard (although they do have broad agreement on the most fundamental categories). For the moment, the IPAS ontologies remain as domain ontologies with no upper level ontology. For any future development however, it will be important to make use of an upper ontology (as one reviewer put it, this is key to avoiding certain well known ontological mistakes).

The representation of change over time has not been fully resolved. There seem to be two main approaches for this issue — three-dimensionalism (3D) and four-dimensionalism (4D) [17]. The 3D approach regards objects as existing only at the present time, whereas the 4D approach regards objects as having a presence that extends through space-time. One 4D upper ontology that has been developed is that of ISO Standard 15926: “Lifecycle Integration of Process Plant Data Including Oil and Gas Production Facilities” [18], which has also been translated into OWL. However, Smith [19] has pointed out its many defects, although these do not necessarily invalidate the 4D ap-

proach. For pragmatic reasons (the IPAS ontologies were developed without the knowledge of these 4D ideas, and there is insufficient time to make further significant revisions before the end of the project), a 3D approach has been taken. So, the composition of generic engines can be represented, as can the current composition of a specific engine. To record which specific parts are in which specific engines, some new classes and individuals will be required — either to represent the state of an engine at a particular time, or (as is currently done) a series of reports of removals and replacements, from which the state at any time can be reconstructed.

5. Conclusions

Here we have tried to collect together some of the lessons that were gained during the process of developing the IPAS ontologies, and in developing the subsequent demonstrators.

Firstly, previous experience and best practice in designing ontologies should be drawn upon, as a bad choice in ontology design can make future updating very difficult. Modularity of ontologies should be planned for from the start. For example, the original Parts ontology was badly designed, as one large ontology rather than separate engines, modules, and components ontologies. Also, the original design choices made it possible to represent a particular engine type, but very difficult to represent additional engines.

Secondly, the "early prototype" approach to building ontologies in this project had the advantage of producing early versions for inspection and use, but also had the disadvantage that the scope of the ontologies was not fully understood while the ontologies were being built.

Thirdly, difficulties were encountered in persuading other partners in the value of ontologies. This may be due to several factors: (i) unclear explanations of ontologies on our part; (ii) the long lead time for developing substantial and stable ontologies contrasting with the quick results expected by others; (iii) confusion as to what the ontologies contained and what functions they could support, due to difficulties in installing and using software such as Protégé.

Finally, the web service implementation of the completed IPAS demonstrators went fairly smoothly, with web services being provided by three of the university partners, and integrated at a one day meeting. This seems to indicate that although the Semantic Web infrastructure may be complicated to set up, it is fairly straightforward to extend.

Acknowledgements

This work is co-funded by the UK Technology Strategy Board's Collaborative Research and Development programme[6]. Our thanks go to the other IPAS partners for their feedback. We also thank the anonymous reviewers for their helpful comments.

[6]http://www.berr.gov.uk/dius/innovation/technologystrategyboard/index.html

References

[1] Rolls-Royce plc. Rolls-Royce TotalCare services webpage. Available at: http://www.rolls-royce.com/service/civil/totalcare/default.jsp.

[2] S. C. Wong, R. M. Crowder, G. B. Wills, and N. R. Shadbolt. Knowledge engineering - from front-line support to preliminary design. In D. F. Brailsford, editor, *Proceedings of ACM Symposium on Document Engineering (DocEng)*, pages 44–52, Amsterdam, The Netherlands., 2006.

[3] M. Gruninger and M. S. Fox. Methodology for the design and evaluation of ontologies. In *Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI*, 1995.

[4] N. F. Noy and D. McGuinness. Ontology development 101: A guide to creating your first ontology. Technical Report KSL-01-05, Knowledge Systems Laboratory, Stanford University, 2000.

[5] M. F. Lopez, A. Gomez-Perez, J. P. Sierra, and A. P. Sierra. Building a chemical ontology using Methontology and the OntologyDesign Environment. *IEEE Intelligent Systems and Their Applications*, 14(1):37–46, 1999.

[6] W3C. OWL Web Ontology Language Guide. Available at: http://www.w3.org/TR/owl-guide/, February 2004.

[7] D. Price and R. Bodington. Applying Semantic Web Technology to the Life Cycle Support of Complex Engineering Assets. In *Proceedings of the ISWC-2004*, pages 812–22. Springer, 2004.

[8] R. Batres, M. West, D. Leal, D. Price, and Y. Naka. An Upper Ontology based on ISO 15926. *Computers & Chemical Engineering*, 31(5–6):519–534, May 2007.

[9] A. Rector, N. Drummond, M. Horridge, J. Rogers, H. Knublauch, R. Stevens, H. Wang, and C. Wroe. OWL Pizzas: Practical experience of teaching OWL-DL: Common errors & common patterns. In E. Motta, N.R. Shadbolt, A. Stutt, and N. Gibbins, editors, *Engineering Knowledge in the Age of the SemanticWeb: 14th International Conference, EKAW 2004*, pages 63–81. Springer, October 2004.

[10] CO-ODE. OWLDoc Home Page. Available at: http://www.co-ode.org/downloads/owldoc/.

[11] A. Gangemi. Ontology design patterns for semantic web content. In *Proceedings of the Fourth International Semantic Web Conference (ISWC-05)*, 2005.

[12] M. Egana, R. Stevens, and E. Antezana. Ontology design patterns for bio-ontologies. In *Bio-Ontologies Meeting, ISMB 2007*, Vienna, Austria, July 2007.

[13] W3C. Representing classes as property values on the semantic web. Available at: http://www.w3.org/TR/swbp-classes-as-values/, April 2005.

[14] W3C. Simple Knowledge Organisation Systems (SKOS) - Home Page. Available at: http://www.w3.org/2004/02/skos/.

[15] Q. Reul, D. H. Sleeman, and D. W. Fowler. CleOn in IPAS: Resolution of lexically incoherent ontologies. Technical Report (forthcoming), University of Aberdeen, 2008.

[16] J. Z. Pan and E. Thomas. Approximating OWL-DL Ontologies. In *the Proc. of the 22nd National Conference on Artificial Intelligence (AAAI-07)*, pages 1434–1439, 2007.

[17] T. Sider. *Four-dimensionalism*. Oxford University Press, 2001.

[18] M. West, C. Partridge, and M. Lycett. Enterprise data modelling: Developing an ontology-based framework for the shell downstream business. In *Formal Ontology Meets Industry (FOMI 2006)*, 2006.

[19] B. Smith. Against idiosyncrasy in ontology development. In *Proceedings of the International Conference on Formal Ontology in Information Systems (FOIS 2006)*, Baltimore, Maryland (USA), November 2006.

Formal Ontologies Meet Industry
S. Borgo and L. Lesmo (Eds.)
IOS Press, 2008

Folksonomies meet ontologies in ARSMETEO: from social descriptions of artifacts to emotional concepts

Matteo BALDONI [a], Cristina BAROGLIO [a], András HORVÁTH [a], Viviana PATTI [a,1], Flavio PORTIS [b], Maurizio AVILIA [a] and Pierluigi GRILLO [a]

[a] *Dipartimento di Informatica, Università degli Studi di Torino, Italy*
[b] *Easybit S.r.l., Torino, Italy*

Abstract. This work focusses on bridging between folksonomies, which provide social but mainly flat and unstructured metadata on web resources, and semantic web ontologies, which instead design structured, machine-processable knowledge spaces. The main purpose is to capture emerging semantics in social tagging systems and to overcome the gap between Semantic Web and Web 2.0, by preserving the complementary advantages of social and ontology-driven methods for describing, categorizing and processing web content. As a way to bridge this gap, we propose a method for linking tags from a folksonomy to concepts of an existing ontology, adopting a statistic approach. We have applied the proposed method to the data collected through the art portal Arsmeteo, relating them to the concepts of an OWL ontology of emotions. Intuitively, by our method we try to capture the *latent emotional semantics* of the tags. Some of the artworks in Arsmeteo could be visited in real exhibitions. In order to capture the emotional potential of the tagging activity during the visit, we explored the possibility to enable tagging of artifacts in real spaces, by using Semacode technology.

Keywords. Social Tagging, Ontologies, Emotions, Semantic Web

Introduction

Nowadays, we can observe many different ways to edit, categorize, search, and share Web content but while the scientific community was researching on how to design and realize the next-generation Web, based on *semantic technologies*, the way to use the Web changed in a way which is summarized by the keyword “Web 2.0". Blogs, wikis (like wikipedia[2]), and social tagging systems (like delicious, flickr, youtube[3]) attract the interest of Web users, partly surely due to the simplicity of the required interaction: plug some content, tag a resource, release a note. This user-initiated use of the Web emphasizes a collaborative perception of Web content, which, in turn, inspired researchers to look

[1]Corresponding Author: V. Patti, Dipartimento di Informatica, Università degli Studi di Torino, Corso Svizzera 185, 10149 Torino, Italy; E-mail: patti@di.unito.it.

[2]wikipedia.org

[3]http://del.icio.us, http://www.flickr.com, http://www.youtube.com

for improved access and retrieval strategies. The collective uploading, and annotating behaviour comprises, in fact, important data for retrieving and presenting Web content.

However, despite the popularity of these technologies and its potential as an information source, the automated deduction of the semantics of annotations as well as of created content, is very limited and convincing solutions still need to be discovered. On the other hand, other (more sophisticate and powerful) systems which, by exploiting semantic web technologies, would indeed allow the generation and handling of knowledge, still lack adequate and as-easy user interfaces and are still mainly thought for a machine-to-machine use. In this context, the interest that motivated our work is to investigate possible solutions to fulfill the need of getting structured and machine-processable semantic information about online content, keeping at the same time the ease of use of Web. 2.0 applications, in the specific setting given by *social tagging* systems. In particular, we focus on the issue of integrating the complementary advantages of social (*folksonomic*) and *ontology*-driven methods for describing, categorizing and processing web content.

Folksonomies are a new user-driven approach to organizing information. They can be seen as "collaboratively generated, open-ended labeling systems that enable users of a community to categorise web content using tags" [16]; as such, they have a dynamic nature, evolving in time [12]. Usually tags are freely chosen keywords, rather than words selected from a controlled vocabulary. At the same time, they are *flat* sets and lack the structure that is required by automatic systems for supplying services to their users. Citing Berners-Lee [19]: "as soon as the user requires more complex processing from the machine, folksonomies reveal their weaknesses and semantic representations become necessary" but folksonomies neither allow the use of reasoning techniques nor they support the interoperability of data. Notice that semantic web ontologies show exactly the features that folksonomies lack: (i) they allow to categorize contents by referring to a vocabulary controlled by experts; (ii) metadata provided by using semantic web ontology are machine processable.

In this work, we propose a method for *bridging* between folksonomies (unstructured collections of metadata expressed in an uncontrolled vocabulary) and structured controlled vocabularies like semantic web ontologies, by preserving the advantages of both. We add to the *social tagging layer* a *semantic layer* that enables the automatic reuse of social content (see also Figure 1). As a result, we will gain a kind of content, that is both tagged by users and associated to an ontology; such content, originally annotated by the members of some community, is now mapped into a *machine-readable* knowledge representation format, thus enabling reasoning, and derivation of new knowledge and information. In this way we couple the naturalness of interaction with the user (due to the folksonomic approach) with the advantages of a shared and machine-understandable semantics, which enables the development of services for the users. The bridge between folksonomies and structured knowledge spaces is achieved by aid of *statistic techniques* relying on data about the natural language words corresponding to the tag and to the textual description of the chosen ontology concepts.

Thanks to the bridging, new semantic relations between the tags (and between the tagged contents) can be automatically deduced. Such new relations can be inferred by reasoning on the ontology, and can be exploited for enhancing the *user's experience* in browsing the web, by creating of a (sort of) ontological order on the tag clouds which usually drives the user's navigation in systems based on social tagging.

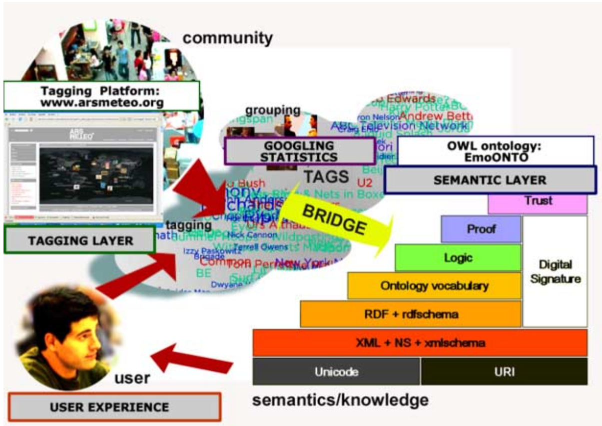

Figure 1. Vision.

Among the application areas that could benefit from this approach there are *art* and *museums*, which share a strong social characterization (people rely on other persons' experience and suggestion). Thus, in this work we applied our ideas to *Arsmeteo*[4], a web application that we have contributed to create. Arsmeteo is a portal for sharing artworks, which allows tagging and tag-based browsing. It enables the collection of digital artifacts (like texts, videos, pictures and music) or digital representations of physical artifacts that are shared by a community, as well as their tagging based on a folksonomic approach. Currently, the portal gathers a collection of about 2000 artifacts produced by over 100 artists. In this framework, the idea is to relate the tags collected by the Arsmeteo platform to concepts of an OWL *ontology* which is particularly relevant in the art domain: an ontology of *emotions*, chosen from the proposals in the literature [9] and adapted to our purpose. Italian words which describe the artifacts, uploaded in Arsmeteo, are linked to Italian emotional words referring to concepts of the ontology. The correlation between *tag-words* and *emotional words* is computed by applying a statistical approach, based on the occurrences (counted by "googling") of the corresponding words in the corpus of the italian Word Wide Web documents. Intuitively, we try to capture the *latent emotional semantics* of the tags.

Last but not least, since some of the artworks in Arsmeteo can be visited in real exhibitions, we have developed a service which allows the visitors of such exhibitions to tag the physical artefacts by exploiting the Semacode technology.

1. Emotional knowledge above the tags in the art domain

Suppose to have a user, Filippo, who is searching and tagging artworks through the Arsmeteo portal. Suppose Filippo's query is "show me pictures related to *happiness*". By linking the tags describing the artworks with concepts from an ontology of emotions, the system could not only find pictures literally tagged with *happiness* but also pictures that

[4]Arsmeteo, http://www.arsmeteo.org/, is inspired by an idea of Giorgio Vaccarino, and is promoted by the *Associatione Culturale Arsmeteo*, which leads and supports the development of the portal.

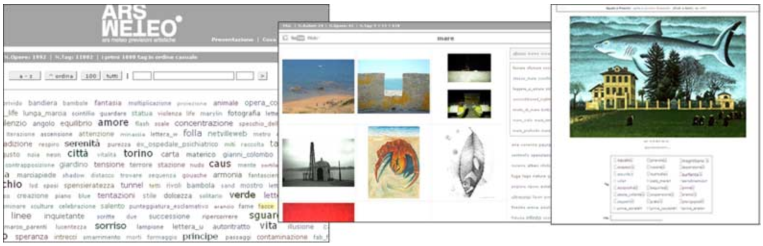

Figure 2. Arsmeteo screenshots. The tag cloud (left); individual tag page for *mare* (center); presentation of a selected resource: preview of the artifact and tagging area (right).

are annotated with other tags which have a latent emotional meaning related to *happiness*. After some ontological reasoning the system could then show further pictures, linked to emotional concepts that are subsumed by *happiness*, e.g. *jubilation* or *enthusiasm*.

Consider now a group of further art associations (museums, art galleries, etc.) which also offer, each through its own web site, the possibility of sharing, searching and tagging artifacts. By relying on the shared knowledge supplied by the emotional ontology, it would be possible to show Filippo also artworks annotated by different communities and belonging to different museums, composing on the fly a personalized exhibition that includes artefacts "lent" by other art portals.

The vision depicted above calls for the definition of three different layers which can be arranged in a circular flow: the social *tagging layer*, the *semantic web layer* and the *user experience layer*, Fig. 1. The contribution of this paper is a method, described in Section 2, for linking the tagging layer to the semantic layer. In the remainder of this section we will describe *Arsmeteo*, the kinds of interaction that it supports, and *OntoEmotions*, an ontology that particularly fits our application purposes, showing how we adapted it to the particular application context. We will, then, briefly introduce the advantages for the system users brought along by the bridging. Such advantages affect content browsing, search and visualization. The actual implementation of these services is the next step to be executed within this project.

1.1. The tagging layer. Arsmeteo: a folksonomic approach to art sharing and tagging

Arsmeteo, on-line since June 2007, enables the collection of digital artifacts (like texts, videos, pictures and music) or digital representations of physical artifacts, that are shared by a community, as well as their tagging based on a folksonomic approach. So far, the community produced a folksonomy of over 10000 tags.

In its core, ARSMETEO is similar to other social resource sharing systems. Registered users can upload multimedia resources and assign arbitrary tags, to them. Once uploaded, artifacts can be browsed and tagged by any visitor of the community. The user can have a preview of the uploaded resources, together with the tags currently assigned to them (see screenshot in Fig. 2(right)). One interesting feature is that a user can also vote the *relevance* of a previously stated tag-artifact relation, by clicking on the plus and minus symbols next to the tag. Such kind of "voting" activity allows the system to associate a *weight* to the tags related to a given artifacts, which will affect the ranking of search results. The tagging activity of the community suggests relations of *similarity* between artifacts, which result somehow categorized based on tags. The set of tags is a

flat namespace: tags are not extracted from predefined sets but they form a folksonomy. With the term *Arsmeteo folksonomy* we refer to the set of terms by which the community of users (artists and visitors) has tagged artifacts.

Artifact search can be performed in a tag-driven way by accessing the search page in Fig. 2(left)), where a tag cloud is used for visualizing the 1000 (or 100) most used tags of our folksonomy. By default the tags appear in random order, their size reflects frequency of use (popularity). Then, it is possible to order the tag cloud both by tag popularity or by alphabetical order. Moreover, searching filters can be applied to select all the tags containing a given string specified as a query in the search input box. When clicking on a tag T of the tag cloud, the user accesses a page (see Fig. 2(center)), whose left side contains previews of the artifacts tagged by T. Such results can be browsed page by page and are ranked, taking into account the relevance voting. On the right side, the user finds a new tag cloud, made of all the other tags related to the retrieved artifacts and tagged by T. Since these new tags describe the same resources tagged by T, they are considered *related* to T. This interface opens the navigation to new unexpected connections with other artifacts, described by the new related tags, but not literally by T (*serendipity*). Moreover, the interface also reports two lists of tags that can be used for refining the search: (1) a list of synonyms for T, and (2) a list of tags with a lexical relation with T (e.g. tags having T as a prefix/postfix, or complex tags containing T).

By clicking on an artifact preview, the user accesses the presentation page for the selected resource A (see Fig. 2(right)), which mainly consists of three parts: the preview of the artifact, the tagging area showing the tags currently describing A, and an area containing a selection of the resources *related to* A. As already mentioned, users can add new tags or they can vote the relevance of the tags already associated to A. Below the tagging area, a list of artifacts *related to* A is presented as a list of thumbnails. Intuitively, such resources were selected the measure of their *connection* to A. The selection is based on tag-similarity, and the related artifacts are ranked, taking into account the relevance voting associated to the describing tags by the community. Thus, the order of the related artifacts in the thumbnail list reflects the degree of connection/similarity calculated by the ranking algorithm. Connections between artifacts are dynamic and change over the time, because of the uploading of new artifacts from the artists and of the tagging and voting activity from the community.

When an artifact is uploaded, besides the tags, the system collects also standard information about the resource like author, genre, year of publication, format. Genres can be described according to a given list of categories. Such information is used by the system to classify contents in standard way and for offering the user access to the repository also by a more traditional kind of search by author, genre and date. Tags can also be added to resources, that are physically shown as part of real exhibitions (e.g. at some art gallery), by means of an application for mobile phones that we have developed and that is described in Section 3.

1.2. The semantic layer: OntoEmotions

One of the most important characteristics of art is that it expresses or stirs emotions. Art can be a record of what the artist is feeling and, at the same time, it can bring about emotional reactions in the viewer. Starting from this consideration, we have chosen to instantiate the methodology sketched in the introduction by linking the Arsmeteo tags

describing artifacts to an ontology of emotions. Even though affective computing has been gaining importance in the last years, there is still no agreement on a standard emotion markup language, complemented by the representation of an ontological structure of emotions. The work of the W3C Emotion Incubator group, that was chartered for defining a general-purpose emotion annotation and representation language, is still at the beginning [14]. We have adopted an emotional ontology, taken from the literature, that met our requirement to have a taxonomic structure mirroring well-founded psychological models of emotions, and that was already implemented by using semantic web technologies: *OntoEmotions* [10,9]. OntoEmotions is based on description logics and has been implemented in the semantic web language OWL. It has successfully been used within a project for developing an emotional voice synthesizer, as an interface between an application for the emotional mark up of text and a voice synthesizer.

OntoEmotions is an ontology of emotional categories, which are structured in a taxonomy covering basic emotions as well as the most specific emotional categories; it includes 85 *concepts*. The basic emotions are: *Sadness*, *Happiness*, *Surprise*, *Fear* and *Anger*. As discussed in [10], the taxonomic structure basically refers to the psychological model by Parrot [17], adapted to these five basic emotions, and integrated with all the emotions which appear in other well-established models. OntoEmotions has been conceived for categorizing emotion-denoting words. So classes corresponding to the emotional concepts have been populated by instances, consisting in emotion-denoting words of two languages: English and Spanish. The ontology has two root concepts: *Emotion* and *Word*. *Emotion* is the root for all the emotional concepts. *Word* is the root for the emotion-denoting words, i.e. the words which each language provides for denoting emotions. In order to allow the classification of words into their corresponding language, the root concept *Word* has two subclasses: *EnglishWord* and *SpanishWord*. Each instance of these two concepts has two parents: one is a concept from the *Emotion* hierarchy (the type of emotion denoted by the word), while the other is a concept from the *Word* hierarchy (e.g. the language of the word). For instance, the word *unhappiness* is both an instance of the concept *Sadness*, and an instance of the concept *EnglishWord*, which means that *unhappiness* is an English word for denoting sadness. Notice that, the class *Emotion* of OntoEmotion has also a special subclass which is called *Neutral*, that in our application can be used for categorizing tags that, according to our measures, do not result to have an emotional potential.

Adapting OntoEmotions to our purposes has been simple. Since the tags used by the Arsmeteo community are mainly Italian words, we have added a new subclass *ItalianWord* to the root concept *Word*, having as instances Italian emotion-denoting words. Our bridging method (between tags and emotional concepts) uses statistic techniques that rely on data, collected by counting the co-occurencies of tag-words and emotion-denoting words in the corpus of Italian Word Wide Web documents. Thus, we needed to add one Italian emotion-denoting word for each emotional concept in the taxonomy. This has been done by using the open source ontology editor Protégé [15]. The list of the Italian emotion-denoting words is the input of the bridging algorithm (Sec.2). The complete list can be found in [5].

1.3. The user experience layer: Tag-based navigation in an emotional space

The link between the tags of the Arsmeteo folksonomy and the concepts of OntoEmotions creates relations and connections among tags (and then among artifacts), opening

the way for the user to experiment tag-based navigation in an *emotional space*. Tags that were unrelated, e.g. *silence* and *donkey*, will be recognized as having a connection to the same emotion, e.g. *sadness* (see table 1). This creates new connections between artifacts tagged with tags which are literally different but are related to the same emotion. New relations can also be created by reasoning on the taxonomic structure of the ontology of the emotions. Notice that, as we will discuss in the next section, a given tag can result as being related in a significant way to more than one emotion, thus providing access to the artifacts, driven by different emotional concepts. Artifacts are usually tagged with many words, that express a variety of meanings and thus support the emergence of different emotional potentials. This is consistent with the idea that art can emotionally affect people in different ways. However, by analyzing the results of the bridging algorithm, we could discover that most of the tags associated to a given artifacts are linked to one particular emotional concept, or to concepts that in the ontology taxonomy are related to one of the basic emotions. For instance, given an Arsmeteo picture tagged only with *ties* and *blood*, by using the results in table 1, that stress a large correlation of both the tags with *fear*, it would be possible to relate the artifact to the *fear* basic emotion.

There are two fundamental challenges at this level: (i) the definition of new methods for sharing retrieving, accessing, and *browsing* content, that take into account the new relations and the semantic emotional structure; (ii) the study of a proper way to visualize to the user the presence of an ontology layer on the tagging layer, with final aim of arriving to a user interface where tags are settled in an emotional space.

(i) **Browsing emotional content**. There are two possible solutions that we would like to explore. *One possibility* is to simply extend the actual Arsmeteo tag-driven search mechanism by offering the possibility to the user to start the search by selecting an emotional category. For example, Filippo could start the search by querying for artifacts related to the emotional concept *sadness*. As result he could get a tag cloud consisting of all the tags of the folksonomy resulting to have an high correlation with *tristezza* (the italian sadness-denoting word). Then, as in normal tag-based navigation, Filippo can choose a tag and access the artifacts described by that tag. Intuitively, the idea is that tags in the tag cloud links Filippo to artifacts that has some tag-driven relation with *sadness*. Moreover, it will be possible to *reason* on the taxonomic structure of emotion ontology in order to offer to Filippo new connections with tags (and then with artifacts). For example, we could show to Filippo tags that can have a high correlation with concepts subsumed by *sadness*, as *nostalgia* and *desolation*. The *second possibility* is to allow the user to search for artifacts that are related to a set of emotions. For instance we could let Filippo search for artifacts that stir contradictory emotions, e.g. *hate*, *fascination* and *rejection*. In this case the system could present to Filippo, rather than a tag cloud, those artifacts which have been described by a list of tags presenting a high correlation with all the three emotions. Again reasoning on the ontology could be used for proposing connections to other artifacts, described by tags with a high correlation to more specialized terns, e.g. *hate-fascination-disgust*, where *disgust* is a concept subsumed by *rejection*.

(ii) **Visualization of the folksonomy in an emotional space**. One interesting issue to study in this context is to find a proper graphic visualization for giving the user the impression to move in an emotional space with an ontological structure when browsing the tags. In this direction it could be interesting to see how to integrate technologies for 3D visualization of ontologies (see for instance the tool OntoSphere [6]) with a 3D version the most popular graphical solution for browsing folksonomies: the tag cloud.

2. Bridging between the Arsmeteo tags and OntoEmotions concepts

Our aim is to determine if there is a relationship between an Arsmeteo tag and an OntoEmotions concept. The first idea that comes into mind is to count the documents in which both the tag and the concept are present. For two reasons, however, this idea is not suitable. First, we look for a method whose result *does not* depend on the frequency of a tag. Second, co-occurrences captures only the very direct relationship when the tag and the concept appear in the same document, while we aim at capturing *latent*, more indirect relationships as well. Hence, instead of counting co-occurrences, we use of a tool of statistics, the *correlation coefficient*.

The study of correlation is performed on the corpus of the Italian Web documents because it is an immense set of everchanging documents reflecting the actual society and its use of language in various ways. The calculation is based on *frequency of appearance* of words which we obtain by Google, a powerful search engine whose advanced interface allows us to perform the necessary queries in automatic manner as described in [5]. In this section we first give some background on “correlation coefficients", then, we show how to apply this concept to look for relationships between Arsmeteo tags and emotions present in OntoEmotions. Finally the application to Arsmeteo is presented.

2.1. Correlation coefficient

Correlation coefficients measure the strength of relationship between random phenomena. Formally, the correlation coefficient of two random variables X and Y is given by:

$$\rho_{X,Y} = \frac{E((X-\mu_X)(Y-\mu_Y))}{\sigma_X \sigma_Y} \tag{1}$$

where E denotes the expected value operator (i.e., it gives the “average” of the random variable given as operand), $\mu_X = E(X)$ is the expected value of X, $\sigma_X = \sqrt{E((X-\mu_X)^2)}$ is the standard deviation of X, while μ_Y and σ_Y are likewise for Y. It is easy to see that $-1 \leq \rho_{X,Y} \leq 1$. To get the intuition behind Def. (1), let us consider an example. If $X = cY$, i.e. Y determines exactly Y, then $\mu_Y = c\mu_X$ and we have:

$$\rho_{X,Y} = \frac{E((X-\mu_X)(cX-c\mu_X))}{\sqrt{E((X-\mu_X)^2)}\sqrt{E((cX-c\mu_X)^2)}} = \frac{E((X-\mu_X)c(X-\mu_X))}{\sqrt{E((X-\mu_X)^2)}\sqrt{E(c^2(X-\mu_X)^2)}} =$$

$$\frac{E(c(X-\mu_X)^2)}{\sqrt{E((X-\mu_X)^2)}\sqrt{c^2E((X-\mu_X)^2))}} = \frac{cE((X-\mu_X)^2)}{cE((X-\mu_X)^2)} = 1$$

If $\mu_Y = -c\mu_X$ then $\rho_{X,Y} = -1$. If X and Y are instead completely independent simple calculations $\rho_{X,Y} = 0$. In a more general setting, if X is large when Y is large and X is small if Y is small then in the numerator of (1) we have a positive value which is as close to the denominator of (1) as strong the relationship is between X and Y. On the contrary, if X is large when Y is small and X is small if Y is large then in the numerator of (1) we have a negative value whose absolute value is as close to the denominator of (1) as strong the relationship is between X and Y.

Several guidelines exist for the interpretation of the correlation coefficient. For the purpose of this work we have found suitable and hence adopted the one proposed in [8]. Accordingly we judge the correlation coefficient as “small” if it is between 0.1 and 0.3, “medium” if it is between 0.3 and 0.5, and “large” if it is between 0.5 and 1.0.

2.2. Correlation between Arsmeteo tags and OntoEmotions concepts

Assume that we are given an Arsmeteo tag and an OntoEmotions concept. To evaluate their relationship we calculate the correlation coefficient of two random variables, X and Y. Given a set of documents, X is the percentage of documents in which the *Arsmeteo tag* is present. Likewise, Y is the percentage of documents in which the *OntoEmotions concept* can be found. In practice, we evaluate the correlation coefficient of X and Y as follows: we identify N disjoint sets of documents of the world wide web by choosing domains; then by simple google queries we determine X and Y for every domain. Let us denote the resulting values of X and Y by $x_i, y_i, 1 \leq i \leq N$. The estimate for the correlation coefficient of X and Y based on $x_i, y_i, 1 \leq i \leq N$ is given by:

$$\frac{N\sum_{i=1}^{N} x_i y_i - \sum_{i=1}^{N} x_i \sum_{i=1}^{N} y_i}{\sqrt{n\sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i\right)^2}\sqrt{n\sum_{i=1}^{N} y_i^2 - \left(\sum_{i=1}^{N} y_i\right)^2}} .$$

2.3. Experiments

The procedure proposed was applied to look for relationships between tags picked up *randomly* from Arsmeteo and the *five principal emotions* present in OntoEmotions which are sadness, fear, anger, happiness and surprise. Since most tags are in Italian, we used the corresponding Italian words which are *tristezza*, *paura*, *rabbia*, *felicità* and *sorpresa*. The disjoint sets of Italian documents were formed by choosing domains corresponding to daily or weekly newspapers (e.g., `repubblica.it`), websites of towns or regions (`intoscana.it`) and webportals providing information on culture or politics (`ex-ibart.it`). Our aim was to look for rather large sets (all the sites contain a few hundred thousand documents) and identify sets which are not limited for what concerns the typology of its documents.

We have performed numerous experiments of which Table 1 reports the correlation coefficient for a few tags that illustrate the insight that one can gain by the proposed procedure. The chosen tags in Italian are *asino*, *uomo*, *centro*, *grattacielo*, *infinito*, *legami*, *sangue* and *silenzio*. Bold numbers indicate large correlation. Let us discuss briefly Table

	donkey	man	centre	skyscraper	infinite	ties	blood	silence
sadness	**0.90**	**0.83**	0.09	0.22	0.20	**0.60**	**0.83**	**0.94**
fear	**0.68**	**0.92**	0.00	0.25	0.11	**0.89**	**0.99**	**0.82**
anger	**0.53**	0.47	0.38	0.29	0.21	0.28	0.42	**0.64**
happiness	0.20	**0.58**	-0.14	0.07	**0.65**	**0.72**	**0.67**	0.34
surprise	0.13	0.16	-0.05	0.29	-0.07	0.09	0.22	0.21

Table 1. Correlation between Arsmeteo tags and emotions.

1. “Donkey” and “man” are living creatures and hence have a lot to do with emotions. The tag “donkey” is associated with four artworks in ARSMETEO and all these works have something disquieting that provoke negative sentiments. The tag “centre” has no large correlation with the five concepts of emotions. It has medium correlation with “anger” which can be caused by its second meaning in politics. This example reveals a general weakness of the method: results for words with *double meaning* are hard to interpret. “Skyscraper” has only small correlation with emotions provoking negative emotions and surprise. The abstract concept “infinite” has large correlation only with “happiness” and

it is interesting to note that many of the works tagged “infinite” in Arsmeteo provoke positive feelings. The Italian word corresponding to “ties” is used primarily to describe emotional ties and hence it makes sense that it has large correlation both with negative emotions and “happiness”. As expected, “blood” has several high correlation values and a very high one for fear. “Silence” is mostly associated with negative emotions while has medium with happiness. Note that significant negative correlation cannot be found in Table 1. This is explained by the fact that the sets of documents are large and contain a high variety of documents and hence it is not probable that the presence of a word implies the absence of another. Negative correlation can be found if we perform the study on smaller set of documents which concentrate on a given subject.

3. Tagging art in real spaces by Semacode technology

Many museums and art galleries have a web site that shows digital reproductions of the artworks that are exposed. The same artists who contribute to Arsmeteo expose their works not only in digital format but also in real exhibitions. Seeing artworks in the setting of an exhibition has a strong emotional impact on the visitors of the museum/gallery, impact that it would be interesting to collect as the result of a tagging activity. Nevertheless, it is quite unlikely that visitors will access the artist’s (or the museum’s) web site, once returned at home, to tag the reproductions of those artworks that impressed them the most, as it is not easy to scatter internet terminals in the exhibition rooms to allow those visitors, who are willing to express their emotions, to tag the reproductions of their preferred artworks. To overcome these limits we have developed a simple application [11], that runs on mobile phones which exploits the 2D-code (more specifically, the *Semacode* libraries for 2D-code generation [4] and the Kaiwa reader [3]) technology to allow the direct tagging of the artworks. Semacodes are two-dimensional barcodes, created on the DataMatrix standard. They are used to encode web page URLs. One of their chief characteristics is that they are easy to read even using cheap optical devices (like mobile phone cameras). Each painting (sculpture, installation) has a 2D-code, which represents a query to a web site, whose execution loads a wap-page which, in turn, allows to tag the artwork, showing at the same time the current tag list for the same object. This application is related to applications like geoblog [2], developed by the “Museo Diffuso della Resistanza" in Torino, that uses 2D-codes to encode information about historical places, that can be read by using a 2D-code reader when visiting such places.

4. Final remarks

In this paper we have proposed a methodology for linking Arsmeteo tags with emotional concepts of the OntoEmotion ontology. Statistical techniques are applied for calculating the correlation between *tag-words* and *emotional words*. The correlation tag-emotion is calculated relying on the occurrences of the corresponding words in the corpus of the Italian Word Wide Web documents. Occurrences are counted by “googling”. Our methodology works properly under the assumption that a precise meaning has been associated to the tag words. Results of correlation with emotional concepts for ambiguous words would be hard to interpret, especially in case of homographs (e.g. *pésca* ’fishing’

and *pèsca* 'fruit'). In order to cope with this problem, we are studying how to enforce our methodology by applying existing NLP techniques for performing homography-level sense-discrimination of tags-words [13] before proceeding to googling. A promising direction to explore is to adopt some WordNet-based query expansion techniques [22]. In particular, we plan to explore the use of the Italian component of the multilingual computational lexicon MultiWordNet [5]. The lexical information stored in MultiWordNet can have a twofold use in our application context. On the one hand, we plan to use it as a pre-defined sense inventory in order to select the most important multiple senses of the Arsmeteo tags and to implement query expansion strategies for capturing the different sense frequencies. On the other hand, we will explore how to use the lexical semantic relations of MultiWordNet for facing the semantic disambiguation problem, e.g. for choosing the sense of a given tag in the context of an Arsmeteo's resource description. Notice that, in our application domain, we cannot count on the traditional notion of context: the only kind of context we can consider is given by the tagged resource and by the other tags used for describing the resource. Moreover, we plan to clean up and optimize the whole process by performing, along the line of [20], a shallow pre-processing on the set of the Arsmeteo tags, aimed at grouping *morphologically similar* tags (e.g. *cane* and *cani*). Once selected a representative for each group of similar tags, we can evaluate the correlation coefficient with emotions only for such tag-representatives.

Folksonomies are *flat* sets and lack the structure that is required by automatic systems for supplying services to their users. Folksonomies do not allow the use of reasoning techniques nor the interoperability of data: as soon as the user requires complex machine processing, folksonomies reveal their weaknesses and semantic representations become necessary. There are on-going researches aimed at giving a structure to a folksonomy by inducing an ontology out of it, e.g. [18]. Other researches aim at understanding how the activity and interactions of many uncoordinated users produces patterns of classification, by exploiting the tools supplied by the study of complex systems [1].

Many other interesting directions could be explored, especially if we consider *specific characteristics* of a certain *application domain*. Museums are a typical example. Organizations like museums have a different role to play in the online world than Flickr, YouTube and the like [7]. Such institutions, by making their collections accessible, aim at providing *knowledge*, rather than *information*. Interpreting artifacts to the general public requires bridging the semantic gap between the professional language of art history and the public perceptions of its visual evidence. The words that a curator uses, in fact, may not be familiar to the average museum visitor. Some important museums (amongst them *Guggenheim*, *Metropolitan Museum of Art*, and *San Francisco Museum of Modern Art*) have been studying for over one year the potential of social tagging in the development of better interfaces, and aim to break the semantic barrier with their visitors, by supporting the project STEVE[6] [21]. Tagging, due to its highly subjective nature, is perceived as a valuable *feedback* that reveals the way in which the public perceives collections, exhibitions, and artifacts. However, this technology is not sufficient: on the one hand, the feedback returned by the visitors in terms of tags cannot be automatically integrated in the museum knowledge base; on the other hand, mere tagging helps in no way the translation of the message/knowledge, that the museum would like to communicate, into

[5] Available at http://multiwordnet.itc.it
[6] http://www.steve.museum/

terms that are more familiar to the public. The integration of semantic technologies is needed.

References

[1] Tagora, semiotic dynamics in online social communities, 2006. http://www.tagora-project.eu/.

[2] Geoblog del museo diffuso della resistenza, 2007. http://memoria.acmos.net/.

[3] The kaiwa reader, 2007. http://reader.kaiwa.com/.

[4] Semacodes, 2007. http://semacode.com/.

[5] M. Avilia. Folksonomie e ontologie in arsmeteo: integrare descrizioni collaborative di opere d'arte con conoscenza ontologica sulla base di indagini statistiche. Laurea triennale in informatica, Corso di Studi in Informatica, Università degli Studi di Torino, 2007. Relatore: Dr.ssa V. Patti.

[6] A. Bosca and D. Bonino. Ontosphere3D: A multidimensional visualization tool for ontologies. In *DEXA '06: Proc. of the 17th Int. Conf. on Database and Expert Systems Applications*, pages 339–343, Washington, DC, USA, 2006. IEEE Computer Society.

[7] J. Seely Brown and P. Duguid. *The Social Life of Information*. Harvard Business School Press, Boston, MA, USA, 2000.

[8] J. Cohen. *Statistical power analysis for the behavioral sciences*. New York: Academic Press, 1969.

[9] V. Francisco, P. Gervás, and F. Peinado. Ontological reasoning to configure emotional voice synthesis. In M. Marchiori, J. Z. Pan, and C. de Sainte Marie, editors, *Web Reasoning and Rule Systems, 1st Int. Conf., RR 2007*, pages 88–102, 2007.

[10] V. Francisco and R. Hervás. Emotag: Automated mark up of affective information in texts. In *Proc. of Doctoral Consortium at the 8th EUROLAN summer school*, 2007.

[11] P. Grillo. Tagging di oggetti reali con il supporto dei semacode. Laurea triennale in informatica, Corso di Studi in Informatica, Università degli Studi di Torino, 2007. Relatore: Prof. M. Baldoni.

[12] T. Hammond, T. Hannay, B. Lund, and J. Scott. Social Bookmarking Tools (I): A General Review. *D-Lib Magazine*, 11(4), 2005.

[13] N. Ide and Y. Wilks. Making sense about sense. In *Word Sense Disambiguation: Algorithms and Applications*. Springer, 2007.

[14] W3C Emotion Markup Language Incubator Group Incubator Activity. http://www.w3.org/2005/incubator/emotion/, 2007.

[15] H. Knublauch, R. W. Fergerson, N. Fridman Noy, and M. A. Musen. The Protégé OWL plugin: An open development environment for semantic web applications. In *Proc. of Int. Semantic Web Conference*, volume 3298 of *LNCS*, pages 229–243. Springer, 2004.

[16] A. Mathes. Folksonomies - cooperative classification and communication through shared metadata. 2004.

[17] W. Parrott. *Emotions in Social Psychology*. Psychology Press, Philadelphia, 2001.

[18] P. Schmitz. Inducing ontology from flickr tags. In *Proc. of the Collaborative Web Tagging Workshop (WWW'06)*, 2006.

[19] N. Shadbolt, T. Berners-Lee, and W. Hall. The semantic web revisited. *Intelligent Systems, IEEE*, 21(3):96–101, 2006.

[20] L. Specia and E. Motta. Integrating folksonomies with the semantic web. In *Proc. of 4th European Semantic Web Conference*, volume 4519 of *LNCS*, pages 624–639. Springer, 2007.

[21] J. Trant and B. Wyman. Investigating social tagging and folksonomy in art museums with steve.museum. In *Proc. of the Collaborative Web Tagging Workshop (WWW'06)*, 2006.

[22] E. M. Voorhees. Query expansion using lexical-semantic relations. In W. B. Croft and C. J. van Rijsbergen, editors, *Proc. of the 17th Annual Int. ACM-SIGIR Conf. on Research and Development in Information Retrieval*, pages 61–69, 1994.

Formal Ontologies Meet Industry
S. Borgo and L. Lesmo (Eds.)
IOS Press, 2008

Relationship Discovery Ontology in Asymmetric Warfare

Kofi Apenyo
U.S. Army Research Laboratory

Abstract: Nowadays, with battles increasingly being fought in urban theaters, enemy combatants can include not only the military but also pseudo-military forces and the civilian population. Given the new battlespace and the realities of the Information Age, military intelligence needs to be processed in new ways in order "to know the enemy". Just as data models and information systems such as the Joint Consultation Command & Control Information Exchange Data Model and the Blue Force Situational Awareness System provide support for the traditional military, novel information systems that capture enemy force and command structure must be built and utilized to achieve information superiority in asymmetric warfare. An information system with a core of the relationships among humans, relationships among organizations, and relationships among humans and organizations is postulated to improve the commander's understanding of enemy presence in his area of operations. Accordingly, this paper develops a relationship discovery ontology from basic modeling principles and then presents a prototype system built on the Protégé ontology engine. The prototype, populated with synthesized but representative human intelligence messages, has been used to demonstrate that a commander may obtain actionable intelligence to capture adversaries and preempt potential enemy attacks. The author validated the ontology model by comparison with concepts from other technologies, by mapping human intelligence messages to model concepts and properties, and by demonstrating the functionality of the prototype.

Keywords: asymmetric warfare, knowledge modeling, ontology, relationship discovery

Introduction

The battlespace of the 21st century is characterized to a large extent by the realities of the Information Age (1). Information has greatly expanded the battlespace beyond the traditional physical battlefield; has led to loss of privacy and remoteness with television, computers, and other media; and has altered the nature of the combatants. The work described herein, development of a relationship discovery ontology, is directed toward helping defeat the new breed of combatants. Nowadays, with battles increasingly being fought in urban theaters, combatants can include not only the military but also pseudo-military forces and the civilian population. For example, in Operation Iraqi Freedom (OIF), we have had Saddam's army, al-Qaeda in Iraq, Shiite and Sunni militants, and civilians. It was well-known in past tank-to-tank battles and naval engagements that civilians played supporting roles in areas such as reconnaissance, food supply, and knowledge of the local terrain. Today, civilians are playing a dramatically bigger role in the battlespace. For example, most of the operations in OIF and Operation Enduring Freedom (OEF) are being conducted

basically in the civilian sector where non-military terrorists and insurgents operate as the primary combatants. Often these combatants are organized in cells that may or may not report to higher level command. In contrast to traditional military command structures, other relationships hold sway in today's "asymmetric forces".

Relationships among humans, relationships among organizations, and relationships among humans and organizations are the asymmetric forces' equivalent of force structure and chain of command. Just as data models and information systems such as the Joint Consultation Command & Control Information Exchange Data Model (2) and the Blue Force Situational Awareness System (3) provide support for the traditional military, novel information systems that capture enemy force and command structure must be built and utilized to achieve information superiority in asymmetric warfare. An information system with a core of the relationship types outlined herein will improve the commander's understanding of enemy presence in his area of operations. A relationship discovery service (RDS) is envisioned as just such a system. For example, having captured an insurgent, a commander can input the insurgent's identity into RDS and receive an output of the insurgent's associates and the organizations he is affiliated with. If judiciously chosen concepts are included in the RDS model, the system can be employed to provide actionable intelligence to capture adversaries and preempt potential enemy attacks.

In addition to the commander, military intelligence analysts constitute another important class of potential users of RDS. For the latter's convenience, RDS should be sufficiently symmetric in use so that starting with an arbitrary instance, this user group should be able to navigate the system to track high value individuals (HVI) in RDS.

The remainder of this paper is organized as follows. Section 1 presents a statement of the problem, discusses alternate approaches to solving it, and advocates the use of web ontology. Section 2 constructs an ontology model from modern knowledge modeling principles, including concepts needed to enable actionable intelligence. Section 3 discusses the implementation of a prototype of the RDS ontology and knowledgebase with the Protégé ontology engine. Validation of the RDS ontology is discussed in Section 4, and Section 5 provides conclusions and suggests future directions.

1. Problem Statement

The information systems building task to achieve relationship discovery may be reduced to the following statement:

Build a tool that uses military intelligence to determine a given individual's associations with other individuals and organizations in a battlespace.

We postulate that such a tool will improve the commander's understanding of enemy presence in his area of operations. For example, with the help of the relationship discovery tool, the commander may be able to extract sufficient information from a captured enemy combatant to foil planned attacks.

1.1.The Datasets

The dataset that will drive RDS consists of large volumes of human intelligence (HUMINT) messages gathered from a battlespace of interest. These data reside on military intelligence information systems such as the Distributed Common Ground System-Army (DCGS-A) and public domain ontologies and databases such as Profiles in Terror (4). Nothing in the analysis precludes the use of other sources of information to power RDS, for example, non-HUMINT intelligence data, news, terrorist databases, and fusion products. For model validation, the present effort uses a sample dataset of 100 HUMINT messages provided by Dr. Jeff Simon, CERDEC, Fort Monmouth (5). A sample HUMINT message is *"01/05/07 - Increased hostile sentiment being expressed against U.S. Troops by many worshippers outside the al-Anbia mosque in Adhamiya."*

1.2.Approaches to the Problem

Several approaches to solve the relationship discovery problem have been considered, including text-mining software, social network analysis, database systems, and web ontology.

A plethora of text-mining software (such as NetOwl, Automap, and Inxight) has emerged in recent times, which promises to analyze structured and unstructured text for entities and relationships. Application in the RDS domain involves input of HUMINT messages to a text miner which then categorizes entities and relationships for populating an RDS knowledgebase. Some text miners have pre-set categories that frequently include some of the following entity types:

- People, Organization, Place, Event, and Time

Interestingly, these same entity types appear in the model we have independently derived from basic modeling principles. However, text mining is not the primary technology for the relationship discovery problem, primarily because it does not have comprehensive data management functionality. The technology we seek must have sufficient data management functionality and, for a seamless product, must be native to the web.

Another technology that immediately comes to mind for application to the relationship discovery problem is social network analysis (SNA). A social network is a structure made of nodes that are linked by one or more specific types of relationships, such as values, visions, ideas, financial exchanges, friendship, or kinship. Metrics for the links include Betweenness, Closeness, Clustering Coefficient, and Reach. SNA is routinely used to find individuals and organizations related to a given individual. However, SNA lacks the ability to describe entities fully, provide essential data management functionality, navigate links with a query language, or do searches under program control to "connect the dots," so to speak. Inability to provide actionable intelligence is another drawback of SNA. Krebs (6), who used SNA on 9/11 data, concluded that "social network analysis is applied more successfully to the prosecution, not the prevention, of criminal activities." Prevention of enemy attacks with the use of actionable intelligence is a capability we intend to build into RDS. For this functionality, we must turn to other information technologies.

Data-driven problems such as the one at hand have traditionally been handled with database technology. Database solutions have been used for about four decades, starting with hierarchic structures, to network structures, and then relational systems. For the relationship discovery problem, the main drawback of the database approach is that the technology was developed in the pre-Internet era. Retrofitting to take advantage of the web cannot be expected to yield an elegant solution. Instead, we look to a new data management technique, web ontology, which has many of the advantages of database systems, to address the relationship discovery problem.

What is web ontology? There are numerous definitions in the literature, most of them based on an early definition by Thomas Gruber (7): *A specification of a conceptualization.* We prefer the following definition, attributable to Deborah Nichols and Allan Terry (8), that is tailored to the semantic web vision: *An ontology is a systematic formalization of concepts, definitions, relationships, and rules that captures the semantic content of a domain in a machine-readable format.*

At the leading edge of the data management evolutionary path, web ontology naturally uses concepts first developed for database systems. Having been developed during the era of the Internet, the technology uses web concepts such as universal resource identifier (URI) and namespace concepts as integral elements and therefore manifests naturally and seamlessly in the Internet environment. Because web ontology has been developed to allow an ontology to interoperate with ontologies at other web sites, with databases, and with Web Ontology Language (OWL) files, information can be accessed and integrated from all these sources. A final advantage of a web ontology approach to the relationship discovery problem is that other ontology-based solutions have been given in similar domains such as terrorism, and we stand to benefit from those lessons (4, 10).

2. RDS Ontology

2.1. Conceptual Modeling

Given that the domain and scope of the RDS ontology is a modern asymmetric battlespace, we proceed to develop an ontology based on contemporary knowledge modeling principles in the following. Like OWL, we have used the Unified Modeling Language (UML) building blocks for capturing classes and associations in battlespace relationship discovery.

2.1.1. RDS Classes and Subclasses

In regard to the problem stated in Section 1, it is clear that the two main classes for modeling relationship discovery are Human and Organization, as shown in Figure 1.

The definitions of Human and Organization are conveyed to the implementation by the various datatype properties defined for them. In practice, properties are derived from a process of abstracting classes using relevance and completeness for the domain

of discourse as guidelines. By analogy to the well-established relational systems discipline, datatype properties are attributes. Because it is often useful in knowledge modeling to explicitly specify subclasses for concepts, we have crafted numerous specializations or subclasses for the Human, the Organization, and other battlespace classes. Human, for example, has domain-relevant subclasses such as insurgent and terrorist, among many others. These subclasses, in turn, have their subclasses, and so on. Being object-oriented, subclasses in an ontology inherit properties from their superclasses. In addition, a subclass may have its own specific properties. Excerpts of Human and Organization taxonomies are shown in Figure 2.

2.1.2. Object Properties

In addition to datatype properties, an ontology may have object properties. An object property is a relationship between two, not necessarily distinct, classes. A binary object property relates two distinct classes. For example, binary property *heads* relates Human with Organization, as in "Human *heads* Organization." A recursive object property relates a class to itself as in "Human *supervises* Human." Introduction of three binary object properties (*heads, isMemberOf, isAffiliatedWith*) between the Human and Organization classes, one recursive property (*hasContactWith*) on the Human Class, and one recursive property (*cooperatesWith*) on the Organization class produces the basic RDS ontology model shown in Figure 3. In reality, numerous property types are required to fully capture the classes, relationships, and their semantics in the domain of discourse.

Traditionally, this level of modeling has been called the terminological component. It defines the terms and structure of the domain of discourse. The second part, the assertional component, uses instances or individuals to manifest the terminological definitions. In a Venn diagram in Figure 4, binary properties "isMemberOf" and "heads" connect instances of Human and Organization; recursive property "hasContactWith" connects two Human instances; and recursive property "cooperatesWith" connects two Organization instances.

The simple RDS model in Figure 3 enables queries of the following types:

1. Information about individual humans,
2. Information about a human's associations with other humans,
3. Information about individual organizations,
4. Information about an organization's associations with other organizations,
5. Information about a human's associations with organizations, and
6. Information about an organization's associations with humans.

Figure 1. Primary Classes of RDS Ontology

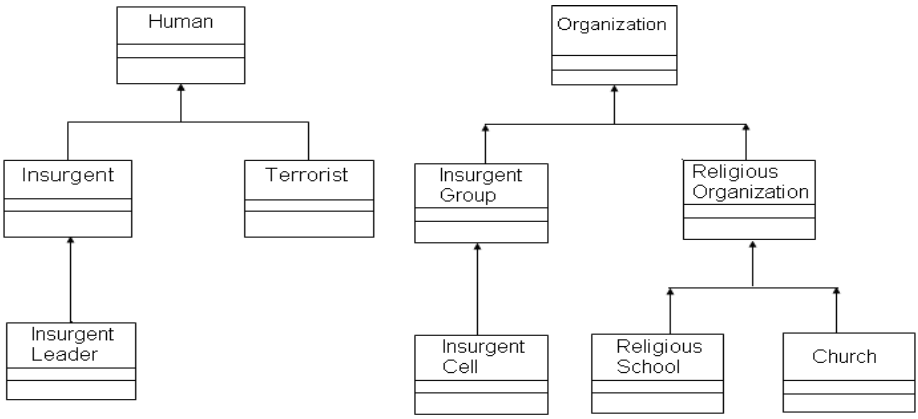

Figure 2. Excerpts of Human and Organization Taxonomies

A knowledgebase built on this model provides the basic tool required for the problem at hand, assuming, of course, that we implement all relevant properties and that supporting HUMINT messages are available online to drive the system. However, the Army transformation requires more than just basic information from its data systems; it demands actionable intelligence for operations in the battlespace (11).

2.2. Extending the RDS Model for Actionable Intelligence

Actionable intelligence may be characterized as having the necessary information immediately available in order to deal with the situation at hand. From the Army's perspective (12), "actionable intelligence refers to a product developed for commanders and Soldiers to provide shared situational understanding allowing commanders and Soldiers to operate with the speed, accuracy, and timeliness necessary to conduct successful operations…"

Integrating actionable intelligence into our model would mean, for example, that we provide information not only about persons but also about where the persons may be found. With location information, the commander is in a position to make a decision to apprehend or attack the persons. Thus, we include a Location class in the RDS model. The Location class is characterized by coordinates, street names, addresses, and various other concepts. Further analysis shows that a recursive property on Location would yield useful information such as nearness, inclusion, and direction. Another building block is an Event class.

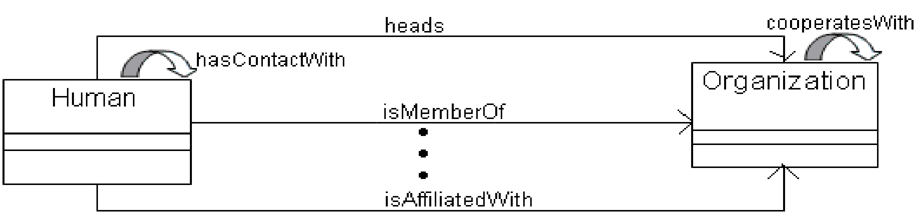

Figure 3. Basic RDS Model with Some Properties

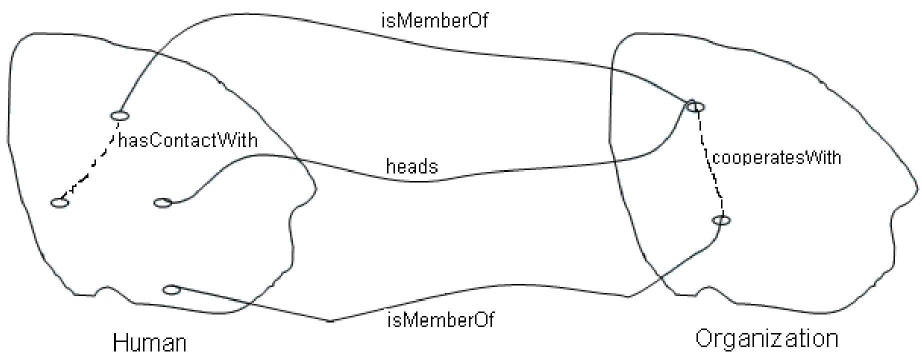

Figure 4. Ontology Instance Diagram

Here, Event stands for an asymmetric warfare incident such as a battle, an attack, an ambush, a political revolution, or an incitement to violence. Inclusion of an Event class gives the commander the ability to find individuals and organizations associated in a variety of ways with a given event. With recursion, Event could be used to find information such as sub-events and other related events. Yet another useful concept for providing actionable intelligence is the resources or assets of humans and organizations. Military assets are characterized by concepts such as Weapon, Transportation Device, and Engineering Component (2). Knowledge of the assets of an enemy organization may be useful for targeting those assets and in anticipating attacks by enemy combatants. For such assets, we define and utilize the Device class, using the Suggested Upper Merged Ontology (SUMO) term (10). Finally, we throw the concept of time into the mix. The Time class being considered here is very general and could be used to capture datatype properties such as Formation Date and Last Attack Date of the Organization class, although we keep the latter for convenience. In RDS, we are particularly interested in identifying events with the time(s) of their occurrence. The resulting RDS model, Figure 5, consists of six primary classes (Human, Organization, Device, Event, Location, and Time), their taxonomies, along with numerous properties and their sub-properties. The RDS model handles a wide range and variety of query types including the set listed in Section 2.1 and is limited only by the number of properties actually implemented.

3. RDS Prototype

A model such as the RDS ontology in Figure 5 is a powerful aid for human understanding of the domain of discourse. To achieve the semantic web vision, however, representation in a language that computers can access and process is required. The World Wide Web Consortium (W3C) recommends and draws attention to the OWL specification and promotes its widespread deployment. Writing OWL is a tedious business; however, editors exist into which one can quickly enter an ontology. These editors can then generate the corresponding OWL file. This effort does not attempt to undertake a formal evaluation and selection process to determine an editor

but rather uses certain technical and industry standards to make the choice. For example, any editor that does not support OWL or the web is summarily eliminated.

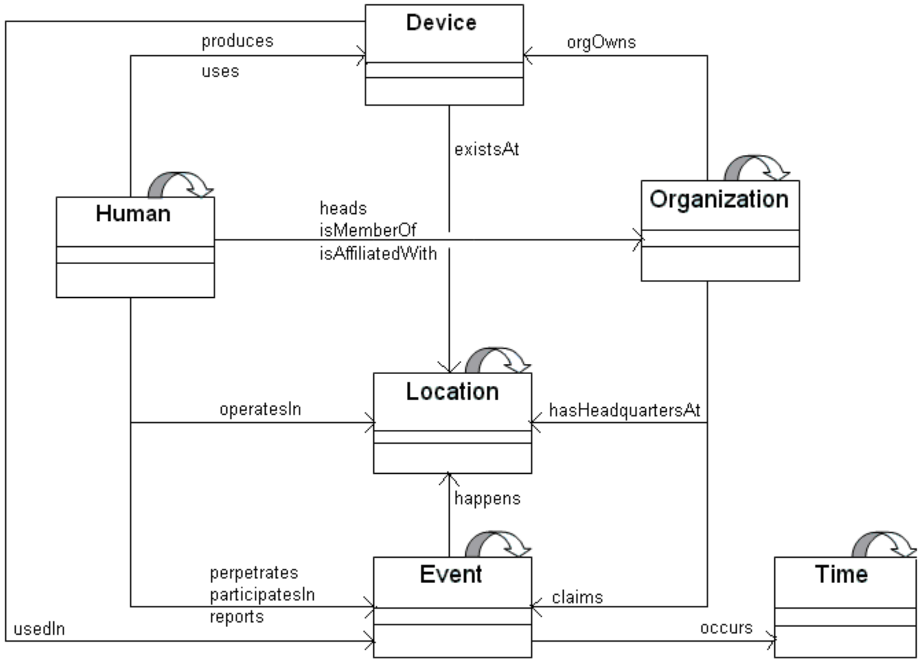

Figure 5. High Level Model of RDS Ontology with Selected Properties

Other criteria employed include (13):

- •Represents a leading edge open source tool with very active development and user communities,
- •Automatically finds inconsistencies and maintains taxonomy and other relationships,
- •Serves as a rapid prototyping environment in which ontologists can instantly create instances and experiment with semantic restrictions,
- •Creates Java classes corresponding to an OWL ontology, and
- •Packages the implementation of applications as plug-ins.

Having examined the survey results of 94 ontology editors (14), we selected Protégé-Owl as the ontology engine for implementing the first RDS prototype. The RDS ontology prototype was implemented on Protégé 3.2.1[1] in a Windows[2] XP environment.

The OWL Classes tab of the Protégé user interface, Figure 6, is used to build the taxonomy of RDS concepts. The Properties tab is used to implement the object and datatype properties of the ontology model. Together, the Classes and Properties tabs are sufficient for creating the RDS schema. During creation of the taxonomy, Protégé

[1]Protégé 3.2.1 is a trademark of Stanford University.
[2]Windows is a trademark of Microsoft Corporation.

flags inconsistencies, keeps track of complex subclass relationships along multiple axes, and exposes hidden relationships. All Protégé environments allow for update and deletion.

In a manner similar to "populating" a database schema with instances to obtain a database, an ontology may be populated with instance data to yield a knowledgebase. A knowledgebase is built on and is a realization of an ontology schema. It stores statements about resources. Thus, we could view a knowledgebase as a collection of

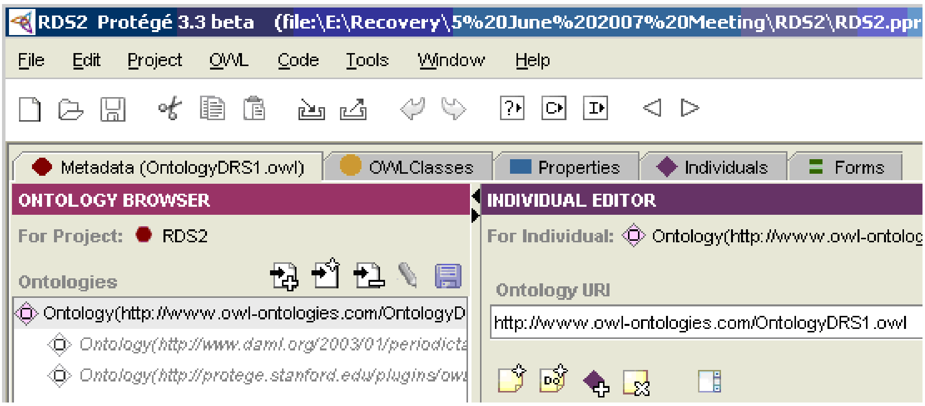

Figure 6. Part of Protégé User Interface

statements about resources. In the case of RDS, the resources are humans, organizations, devices, time, events, locations, and their subclasses. The RDS knowledgebase is built using the Individuals tab shown in Figure 6. With this tab, instances of resources and their relationships are created as prescribed by the schema.

Information in the knowledgebase is stored as statements that are easily accessed through the Protégé forms interface. For example, the statement "Marwin Anour has contactWith Hafiz Waheed" is easily read from the Individuals Editor of Figure 7.

4. RDS Model Validation

Model validation has been defined as "*substantiation that a computerized model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model*" (15). Guided by this characterization, model validation in the RDS context may be established by showing that we have selected the appropriate concepts and integrated them with appropriate properties for discovering relationships in asymmetric battlespaces. To this end, the RDS model validation process was reduced to answering these questions:

1. Do RDS concepts map well to concepts derived for similar domains using closely related technologies?
2. Do representative HUMINT messages map well to RDS concepts and properties?

3.Does a prototype system that implements the model, populated by data mapped from a set of representative HUMINT messages, allow a commander to determine an individual's associations with other individuals and organizations in a battlespace?

We argue that answering all three questions with "Yes" is sufficient validation of the model.

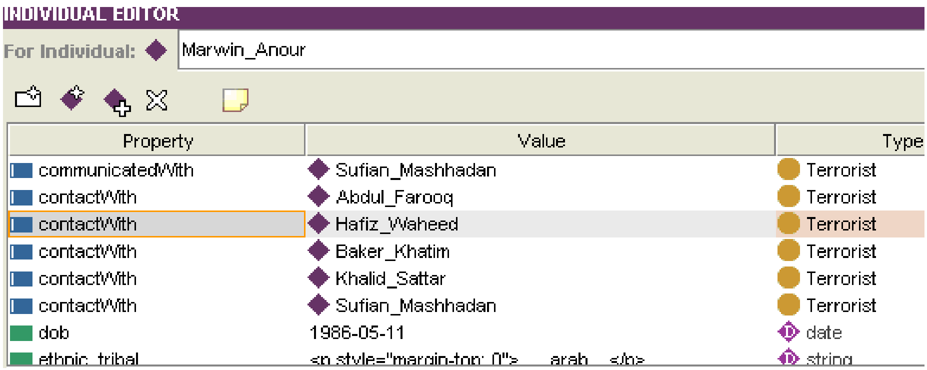

Figure 7. Protégé Individuals Editor

First, we examined related efforts in text mining, SNA, and terrorism information systems for concepts occurring in their data models. We have found that text-mining products have consistently identified the following basic concepts as those that characterize a typical text: People, Organization, Place, Event, and Time. In her SNA work specifically dealing with terrorism (9), Dr. Carley specifies the nodes of a network as People, Units of action, Coalition partners, Departments (Organization), Resources (Assets), Ideas or Skills, Events, Nation-states (Location). In the case of terrorism systems, a number of significant efforts such as the Profiles in Terror project (4) have identified and used, among many others, the following concepts in their database: Person, Organization, Location, Event, Facility, Terrorist, Terrorism Resource. Clearly, RDS concepts map well to concepts used in these three related technologies. We conclude that since research in text mining, SNA, and terrorism that are variously related to battlespace relationship discovery, has consistently developed concepts herein derived for RDS, the classes are valid for this domain of discourse.

Next, we further validated the RDS model by exercising it on data from the domain of discourse. For validation of the ontology, CERDEC created HUMINT messages for an unclassified fictitious Iraqi battlespace scenario (5). To eliminate experimenter's bias, CERDEC was not provided with a copy of the RDS ontology. The set of 100 HUMINT message statements, synthesized but representative of actual classified HUMINT data, was parsed and successfully loaded into the prototype RDS knowledgebase. Validation of the model is established by

1 .each concept in the scenario found its matching class in the prototype,

2. virtually all the HUMINT message properties mapped to prototype properties,
3. RDS was able to ingest all CERDEC scenario data, and
4. all primary RDS prototype classes were populated.

Finally, we addressed the third validation question: "Can an RDS based on this model accomplish the intended purposes?" For the test, a scenario was set up wherein the RDS prototype populated with the data described above is queried. Through a series of queries, a commander successfully navigated the RDS knowledgebase to obtain actionable intelligence to foil a planned attack on a hospital and a primary school in the Iraqi battlespace. The search path also yielded actionable intelligence to order the arrest of a bomb maker in Karachi, Pakistan.

5. Conclusion and Future Work

In this project, we have developed an ontology model for relationship discovery in a battlespace, implemented a prototype of the ontology and its knowledgebase, and validated the model. Much work remains to be performed toward achieving a seamless, powerful relationship discovery tool with comprehensive functionality for the commander in a battlespace.

First, loading HUMINT messages into the knowledgebase is currently a manual process, wherein HUMINT messages are converted into triples then inserted into the knowledgebase one instance at a time. Because of the anticipated large volumes of data, the loading process must be, at least, semi-automated. Text mining may aid this effort. Second, the interface to RDS at the present time is through forms. However, forms interfaces are highly constrained. For more flexibility, Protégé-Owl offers a query language capability, SPARQL. We plan to store commonly occurring queries that can be easily accessed by users. However, for maximum flexibility, we will use a high-level language such as Java into which we embed SPARQL commands. Third, because of the distributed nature of intelligence data systems, we intend to develop interoperability hooks to exploit remote intelligence information to allow the commander to "connect the dots." Fourth, we plan to include reasoning into the RDS ontology so that, for example, if a relative of a member of an organization has contact with the head of the organization, then RDS may reason that the relative is also a member of the organization. Finally, we will explore the possibility of extending SUMO with RDS. Toward this end, we have been careful to use SUMO modeling primitives so that the mapping from RDS to SUMO will be obvious. Extending SUMO will give RDS access to SUMO's extensive taxonomy, definitions, and axioms.

Acknowledgments

I wish to thank Dr. Windell Ingram, U.S. Army Research Laboratory, for helpful feedback during discussions of versions of this paper.

References

[1] Alberts, David S., John J. Garstka, and Frederick P. Stein. *Net Centric Warfare 2nd Edition (Revised).* CCRP Publication Series. July 2002.
[2] Multilateral Interoperability Programme. *JC3IEDM*. http://www.mip-site.org/home.htm
[3] Dittmer, Kurt. "Blue Force Tracking: a subset of combat identification." *Military Review*, 9/1/2004.
[4] Profiles In Terror. http://profilesinterror.mindswap.org/people/
[5] Simon, Jeff. "Humint Messages." Private Communications, CERDEC, 15 Dec 2006.
[6] Krebs, Valdis. "Uncloaking Terrorist Networks." 1st Monday, vol7, no4, Apr '02
[7] Gruber, Thomas R. A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199-220, 1993.
[8] Nichols, Deborah and Allan Terry. *User's Guide to Teknowledge Ontologies.* Teknowledge Corp., December 3, 2003.
[9] Carley, Kathleen M. "Dynamic Network Analysis - Increasing Context Sensitivity." US Army Research Laboratory, 9 May 2007.
[10] Niles, I., and A. Pease. "Towards a Standard Upper Ontology." *Proc. of the 2nd International Conference on Formal Ontology in Information Systems*, Chris Welty and Barry Smith, eds, Ogunquit, Maine, Oct 17-19, 2001.
[11] Iwicki, Stephen K. "Introducing the concept of Actionable Intelligence." *Military Intelligence Professional Bulletin*, Jan-March, 2004.
[12] Burlas, Joe. "Actionable Intelligence relies on every Soldier." *Army News Service*, April 13, 2004.
[13] Knaublauch, Holger. "Ontology-Driven Software Development in the Context of the Semantic Web." Stanford Medical Informatics, Stanford Univ, CA, 2004.
[14] Denny, Michael. "Ontology Building: A Survey of Editing Tools." November 06, 2002. www.xml.com/2002/11/06/Ontology_Editor_Survey.html
[15] Schlesinger, S., et al. "Terminology for Model Credibility." *Simulation* 32, 103-104, 1979.

This page intentionally left blank

Author Index

This page intentionally left blank